Universidad Autónoma de Madrid
Escuela politécnica superior

Grado en Ingeniería Informática

Trabajo Fin de Grado

ESTUDIO DE VIABILIDAD PARA LA IDENTIFICACIÓN DE HOJAS MEDIANTE REDES NEURONALES.

Autor: Alejandro Garo Hernández
Tutor: Marcos Escudero Viñolo
Ponente: Jesús Bescós Cano

Julio 2020

ESTUDIO DE VIABILIDAD PARA LA IDENTIFICACIÓN DE HOJAS MEDIANTE REDES NEURONALES.

Autor: Alejandro Garo Hernández
Tutor: Marcos Escudero Viñolo
Ponente: Jesús Bescós Cano

Video Processing and Understanding Lab. (VPULab)
Dpto. de Tecnología Electrónica y de las Comunicaciones
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Julio 2020

Resumen

Resumen

La identificación y clasificación de imágenes ha sufrido un auge estos últimos años debido a la revolución causada por el Aprendizaje Profundo, capaz de realizar tareas de reconocimiento de imágenes o seguimiento de objetos en vídeo en vivo entre otros, con unos resultados espectaculares.

Con este estudio queremos demostrar, mediante la utilización de un dataset compuesto por imágenes de hojas, la viabilidad de la extracción de características y su posterior clasificación de objetos cuyas características difieren poco. Para la extracción de características utilizaremos una Red Neuronal Residual pre-entrenada y para la clasificación una SVM. Se realizará también una segmentación del dataset previa a su caracterización, para evaluar la potencial mejora introducida por este preprocesado. Visualizaremos el comportamiento de las capas de la red, con la intención de ver su desempeño en cuanto a las características extraídas y analizaremos sus resultados al clasificarlas, las cuales arrojan un buen porcentaje de acierto usando las características de las últimas capas de la red.

Palabras Clave

Aprendizaje Profundo, Visión por Computador, Redes Neuronales Residuales, Segmentación, Extracción de Características.

Abstract

The image identification and classification had great improvements through the last years caused of the Deep Learning *revolution*, capable of performing tasks in image recognition or object tracking in live video among others, with great results.

The purpose of this study is to demonstrate the feasibility of the feature extraction and its subsequent classification of images whose features differ little. For the extraction of characteristics we will use a pre-trained Residual Neural Network and for the classification an SVM. A segmentation of the dataset will also be carried out prior to its characterization, to evaluate the potential improvement performed by this preprocessing. We will analyze the behaviour of the network layers, with the purpose of see its performance in terms of the extracted features and we will analyze their results when classifying them, which give a good percentage of success using the characteristics of the last layers of the network.

Key words

Deep Learning, Computer Vision, Residual Neural Networks, Segmentation, Feature Extraction.

Agradecimientos

A mi familia, por su constante apoyo y por brindarme esta oportunidad.

A mis compañeros, por el viaje compartido.

A mi tutor del TFG, por la guía, ayuda y consejos.

A todo el personal de la EPS, por todos los servicios prestados, es especial a Diego y a Lorenzo.

Índice general

Índice de Figuras

Índice de Tablas

1 Introducción

1.1. Motivación del proyecto

La clasificación de hojas es una cuestión ampliamente estudiada en el ámbito de la Botánica, siendo necesario tener conocimiento de los campos de la Taxonomía y la Filogenia. La clasificación se realiza manualmente mediante Claves Dicotómicas, donde se aplica ese conocimiento para establecer características discriminatorias y así llegar a un resultado, llegando a veces a ser un proceso costoso. Con esto, no es tarea sencilla la identificación de una hoja sin previo conocimiento de esas áreas.

Con el reciente auge y avances en varios ámbitos de la Informática como son las Redes Neuronales Convolucionales y su potencia para la extracción de caracterísiticas, la Visión por Computador con sus métodos para trabajar con imágenes y el Aprendizaje Automático con la clasificación que nos aportan sus algoritmos, en este trabajo se hace un estudio confluyendo estos tres campos, en el cual no se trata de distinguir entre un coche y una bicicleta, si no entre muestras cuyas características a veces difieren por muy poco como son las hojas de árboles..

1.2. Objetivos y enfoque

El trabajo consistirá en:

- Una documentación de los campos que abarca el estudio.
- La elección de una base de datos pública viable.
- La construcción de un sistema de clasificación supervisada con el cual realizar un sistema clásico de clasificación de imágenes

Por otro lado, el sistema implementado consistirá en:

- Un preprocesamiento de la imagen.
- Un entrenamiento de la Red Neuronal con nuestras imágenes.

- Una extracción de las características una vez entrenada.
- Una clasificación utilizando un algoritmo de Aprendizaje Automático, el cual se entrenará con las características extraídas en el paso de entrenamiento para posteriormente clasificar las características arrojadas por la Red Neuronal
- Una comparativa entre los resultados de distintos puntos de la Red así como de las muestras preprocesadas y sin preprocesar.

Se quiere comprobar la eficacia de la extracción de características en diferentes puntos de la red con muestras cuya clase en algunos casos difiere por poco, y el rendimiento de su clasificación.

1.3. Organización de la memoria

La organización de la memoria Se divide en los siguientes bloques:

Capítulo 1. **Introducción:** Motivación del proyecto. Objetivos y Enfoque. Organización de la memoria

Capítulo 2. **Detección de Hojas. Estado del arte:** Introducción. Técnicas existentes para la clasificación de hojas. Bases de Datos Existentes. Estudios Realizados. Segmentación y Detección de Bordes. Redes Neuronales Convolucionales para Extracción de Características. Clasificación y Regresión.

Capítulo 3. **Diseño y desarrollo:** Esquema General del Algoritmo Propuesto. Dataset. Segmentación de la Hoja. Extracción de Características. Clasificación. Comparación de Resultados.

Capítulo 4. **Experimentos Realizados y Resultados:** Descripción de los Experimentos. Resultados y Discusión.

Capítulo 5. **Conclusiones.** Donde se resume el trabajo realizado y se discuten los principales descubrimientos y enseñanzas adquiridas.

2

Detección de Hojas. Estado del arte

2.1. Introducción.

La detección, clasificación y reconocimiento de imágenes es algo ampliamente estudiado, sobre todo en los últimos años con el auge de un campo de la Inteligencia Artificial, el Aprendizaje Automático, confluyendo con otro campo, el de la Visión por Computador. Es en concreto otro campo más específico dentro del Aprendizaje Automático el que ha desempeñado un gran papel en la detección de imágenes, el del Aprendizaje Profundo. Gracias a las investigaciones de las últimas décadas en Aprendizaje Profundo, junto con herramientas facilitadas por la Visión por Computador, se han obtenido resultados prometedores en esta cuestión.

El *Aprendizaje Automático* es el encargado, mediante algoritmos y datos, de inferir conclusiones sobre dichos datos, mientras que el *Aprendizaje Profundo* son una serie de algoritmos basados en redes neuronales, un subtipo de algoritmo dentro del Aprendizaje Automático, los cuales admiten datos matriciales. Esto es perfecto para poder procesar imágenes, dado que en definitiva su representación computacional es la de matrices numéricas multidimensionales.

La *Visión por Computador* nos permite mediante una serie de herramientas, obtener, analizar y procesar imágenes de manera que un ordenador pueda trabajar con ellas.

Debido a los avances que ha tenido los últimos años, cada vez más está siendo introducida en el la vida cotidiana de las personas. Un ejemplo sería el el control biométrico de las huellas dactilares, usado como manera prácticamente inequívoca de acceso a un área restringida.

Para los ojos y cerebro humanos, detectar, analizar y reconocer un objeto es algo banal, es intrínseco aprender y reconocer objetos, aún pudiendo estos objetos variar en infinidad de formas, pero para un computador al contrario, es algo muy complejo. La dificultad reside en que para un computador, como se ha dicho antes, una imagen es una matriz numérica tridimensional, entonces para encontrar semejanza entre objetos tiene que encontrar patrones en dichas matrices.

2.2. Técnicas existentes para la clasificación de hojas.

Las técnicas existentes para la clasificación de hojas difieren tanto en el campo desde el cual se aborden como las técnicas a utilizar. Históricamente, la identificación de hojas no ha sido fácil, dada la enorme cantidad de especies que las componen y las técnicas usadas. Hoy, es un tema que se puede abordar desde el campo de la computación, haciéndolo más accesible, rápido y eficaz.

2.2.1. Biología

El campo de la clasificación de hojas corresponde a la biología, más concretamente al campo de la *Filogenia* y la *Taxonomía*.

La ***Filogenia*** es la parte de la biología que se ocupa de las relaciones de parentesco entre los distintos grupos de seres vivos [12].

En este campo, la Filogenia se encuentra fuertemente relacionada con la ***Taxonomía***, la cual es la ciencia que se encarga de la clasificación de organismos [13]. Esta clasificación debe ser congruente con la clasificación filogenética, aunque cada escuela filogenética utiliza su manera de clasificación, esto es, según la escuela fenética, la cladística y la sistemática evolutiva [14].

La filogenia y la taxonomía utilizan para la identificación de organismos las llamadas *Claves Dicotómicas*.

Claves Dicotómicas

Son unas método de identificación de especies las cuales segun las características de organismos realizan una clasificación [15].

Constan de un conjunto de descripciones de dichos organismos, excluyentes entre sí, permitiendo identificarlos mediante las sucesivas elecciones de las descripciones, eligiendo una en cada paso [15]. En cada elección hay dos opciones. Un ejemplo sería la Figura 2.1.

CLAVE DICOTÓMICA

Paso 1)	Presencia de hojas	Paso 2
	Ausencia de hojas	Paso 10
Paso 2)	Tallo leñoso	Paso 3
	Tallo flexible	Paso 7
Paso 3)	Árbol	Paso 4
	No árbol	Paso 5
Paso 4)	Hoja perenne	**EJEMPLO 10 (PINO)**
	Hoja caduca	**EJEMPLO 5 (ROBLE)**
Paso 5)	Espinas	**EJEMPLO 3 (ZARZAMORA)**
	Sin espinas	Paso 6
Paso 6)	Hoja aguja	**EJEMPLO 6 (ENEBRO)**
	Hoja no aguja	**EJEMPLO 1 (HELECHO)**
Paso 7)	Hoja ovada	**EJEMPLO 2 (ORÉGANO)**
	Hoja no ovada	Paso 8

Figura 2.1: Porción de una Clave Dicotómica. Extraído de [1]

Es importante conocer el organismo a identificar mediante una Clave Dicotómica para poder elaborar correctamente estas descripciones.

Dichas claves son dependientes de lo que se va a clasificar y del autor, aunque siempre se tendrán en cuenta unas características fijas, pudiendo variar en la posición de las ramas.

2.2.2. Informática

Utilizar técnicas informáticas para la clasificación de hojas puede abordarse simulando las *Claves Dicotómicas* o mediante algoritmos de *Visión por Computador* e *Inteligencia Artificial*.

En el ámbito de la ***Informática***, una Clave Dicotómica, puede entenderse como una representación en un árbol binario o grafo no completo, con información en sus conexiones, donde utilizando un algoritmo de Búsqueda Informada se podría resolver en base a comparaciones y diferentes estrategias para determinar el orden de búsqueda.

Hay que añadir el hecho de construirlo, ya que se debe hacer a mano y se requieren conocimientos de taxonomía y botánica, en este caso concreto, y un mantenimiento de la clave, dado que si se añade una nueva especie se tendría que modificar el árbol acorde a la correcta clasificación de la nueva hoja.

Se puede abordar también utilizando técnicas de clasificación de imágenes, combinando técnicas de *Visión por Computador* e *Inteligencia Artifical*.

La ***Visión por Computador*** nos ofrece técnicas de tratamiento, procesamiento y análisis de imágenes, con el fin de que puedan ser entendidas por un ordenador, intentando simular la forma la cual un ser humano entiende una imagen. Por otro lado, la Inteligencia Artificial, según Bruno López Takeyas [16], *Es una rama de las ciencias computacionales encargada de estudiar modelos de cómputo capaces de realizar actividades propias de los seres humanos en base a dos características primoridales: el razonamiento y la conducta.* En concreto, la clasificación de imágenes se resuelve mediante técnicas de ***Aprendizaje Profundo***, el cual permite que los modelos computaciones que se componen de múltiples capas de procesamiento (Redes Neuronales) aprendan representaciones de datos con múltiples niveles de abstracción [17].

Para llevar a cabo la clasificación de imágenes se deben seguir, dependiendo del sistema, una serie de factores que son, la elección de una colección válida de datos para entrenar el clasificador en el caso de sistemas supervisados, un sistema de clasificación válido, técnicas de preprocesamiento de la imagen, técnicas para la extracción y elección de características, procesamiento post-clasificación y métodos para precisar los resultados obtenidos. Todos estos factores son determinantes para conseguir buenos resultados.

Elección del Clasificador.

La elección del clasificador comienza eligiendo entre *Clasificación Supervisada* y *No Supervisada*. Para ello se describe primero que son los Clasificadores Supervisados y No Supervisados y una justificación de los datos a elegir.

- **Datos de entrenamiento en clasificadores supervisados:** los Clasificadores Supervisados son aquellos que necesitan un conjunto de datos ya etiquetados para tener resultados. Para la clasificación supervisada de imágenes es crítico conseguir un alto número de datos, pues sin ellos, el entrenamiento del clasificador podría dar resultados erróneos en la clasificación. Estas imágenes deben ser acordes a las cuales se van a clasificar. El propósito general de los clasificadores supervisados es el de hacer predicciones, ya que teniendo un conjunto de datos ya clasificados, se puede inferir a qué clase pertenecen los nuevos datos, o el de hacer una regresión, determinando el comportamiento de los datos a partir de los ya clasificados [18].

- **Datos de entrenamiento en clasificadores no supervisados:** los clasificadores no supervisados no necesitan de datos ya etiquetados para la clasificación, en cambio, se basa en las características de los datos para así agruparlos y ver similitud entre ellos. Para la clasificación no supervisada de imágenes se necesita hacer una elección correcta y no muy extensa de las características a analizar de los datos, dado que la clasificación se hará correspondiendo a la similitud que haya entre las mismas. Este tipo de clasificador tiene un propósito exploratorio, ya que no se cuenta con datos comprobados sobre los que entrenar los nuevos datos, pero se puede describir la estructura de los datos e intentar encontrar algún tipo de de organización entre ellos [18].

La elección del clasificador entonces se resuelve entonces según el propósito el cual queramos hacer con nuestros datos, la cantidad de datos de los que dispongamos y si estos datos están o no etiquetados.

En el campo de la clasificación de imágenes tenemos que tener también en cuenta la cantidad de datos que tengamos, la calidad de los mismos, diferentes datos, disponibilidad de algoritmos de procesamiento de imágenes y de clasificación y la necesidad del usuario [19].

Preprocesamiento de la Imagen.

Se engloba dentro del campo del *Procesamiento Digital de Imágenes*, el cual es el conjunto de técnicas que, aplicadas a las imágenes, buscan mejorar la calidad o la búsqueda de información en ellas [20]. Utilizando este tipo de técnicas adecuamos nuestro conjunto de datos con el fin de conseguir un mejor análisis, consiguiendo nuevas imágenes a partir de las procesadas. A éstas técnicas, cuando se aplican a nivel local, se les denomina filtros.

El conjunto de técnicas y filtros es muy diverso, y sus principales usos son:

- **Conversión a escala de grises:** es equivalente a obtener la luminancia de la imagen. El ojo humano percibe la intensidad lumínica de un objeto según el color, y por ello, para obtener la escala de grises de una imagen se ha de hacer una media ponderada de cada color, según los criterios deseados [21]. La Figura 2.2a muestra un ejemplo.

- **Cambio en el contraste:** es una de las técnicas más significativa cuando se hace preprocesamiento de imágenes. El contraste es la diferencia de luminancia reflejada por dos superficies adyacentes. Se pueden distinguir principalmente dos tipos de contraste, el contraste tonal y el de color. El contraste tonal se produce entre las sombras y las luces y el total de la tonalidad de grises. El contraste de color es aquel producido por combinar distintos tipos de tonalidades de color en una misma imagen, afectando al modo el cual se percibe la imagen [22]. En la Figura 2.2b se muestra una imagen con distintos contrastes.

Con esto, podemos conseguir resaltar características de nuestra imagen o corregir la tonalidad de color o su iluminación.

- **Eliminación del ruido:** El ruido se define como como la variación aleatoria del brillo o el color en las imágenes, dependientes del dispositivo de entrada. Se puede dar tanto en imágenes obtenidas mediante un dispositivo digital como uno analógico. La razón de que se produzca se debe a que el sensor de la cámara o bien no recoge la información a capturar de manera adecuada y el procesador tiene que interpretarla o debido a que el sensor esté sobrecargado y genere información aleatoria [23]. En la Figura 2.2c se muestra una reducción de ruido.

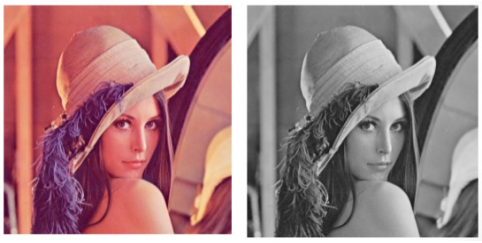
- **Realzamiento y detección de bordes:** facilita información acerca del contorno y límite del objeto, obviando su contenido, útil en cuanto al análisis y detección de objetos y para aplicaciones de filtrado. Como se centra en detectar y realzar contornos, reduce significativamente la cantidad de datos a procesar, haciendo más liviano y eficiente el análisis de imágenes [24].

Los bordes del objeto se consiguen detectando y realzando los cambios bruscos en los píxeles vecinos existentes en una imagen mediante el uso de filtros específicos.

Según se aplique un filtro u otro se pueden obtener distintos resultados, algunos consiguen obtener puntos o contornos de detalles de la imagen o sólo los contornos más obvios.

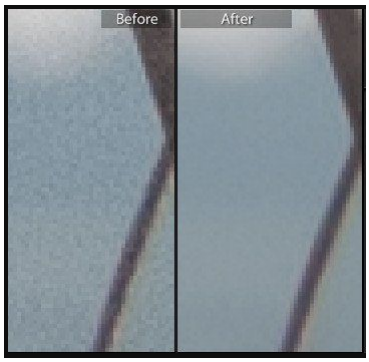
Los bordes son invariantes a los cambios de luz, con lo cual se puede obtener una normalización entre un dataset de imágenes, pero son muy sensibles al ruido, por lo que conviene aplicar filtros de suavizado.

La Figura 2.2d recoge un ejemplo de detección de bordes.

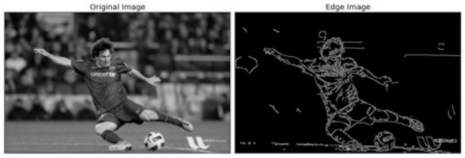


(a) Conversión a escala de grises. Adaptada de [25]

(b) Cambio en el contraste. Adaptada de [26]



(c) Reducción de ruido. Adaptada de [27]



(d) Detección de bordes. Adaptada de [28]

Figura 2.2: Métodos de Preprocesamiento

Visto el conjunto de técnicas explicadas, conviene estudiar cada caso de manera independiente, pues cada imagen puede ser distinta a otra y por ello necesitar aplicar un conjunto de técnicas de preprocesamiento distintas, aunque bien es cierto que hay unas técnicas más recomendables a otras según el caso al que nos enfrentemos.

Extracción y Elección de Características.

La *Extracción de Características* en imágenes es un campo inmenso el cual está en investigación y constante cambio. En él confluyen áreas como son la informática, las matemáticas, la física y la fotogrametría, entre otros. Estas investigaciones abarcan desde la aplicación de filtros lineales como la aplicación de algoritmos de aprendizaje profundo, que pueden entenderse como la aplicación sucesiva de filtros introduciendo relaciones no lineales. Dada la cantidad de métodos existentes, no hay ninguno que sea *universal*, más bien los requerimientos del problema dictarán qué método se utilizará y si es necesario personalizarlo.

Muchas veces las imágenes contienen información que proporcionan poca información o son irrelevantes para el análisis, siendo tarea del desarrollador el aplicar técnicas que las supriman lo mayor posible, la elección de un algoritmo adecuado y la colección de un dataset válido.

La extracción y selección de las características correctas es un paso decisivo para la clasificación de imágenes, ya que se utilizan para distinguirlas. Se pueden categorizar según los componentes clave de la imagen, que pueden ser, entre otros, la intensidad del color, la textura, los bordes de los objetos o su forma. La eficiencia de la extracción de características mejora la clasificación de la imagen, con lo cual se requiere una extracción de características compacta y relevante. Se pueden distinguir dos grupos de técnicas de extracción de características, las de alto nivel y las de bajo nivel [29]:

- **Técnicas de bajo nivel:** consisten en los pequeños detalles de la imagen, como los puntos, las líneas, las esquinas o curvas del contorno de los objectos, etc. Estas características se pueden extraer de la imagen sin tener conocimiento previo sobre ella.

- **Técnicas de alto nivel:** son aquellos que se construyen sobre las características de bajo nivel para detectar objetos y formas en las imágenes. Por ejemplo, en el campo de la clasificación de hojas, después de haber detectado el contorno de la hoja, si tiene bordes serrados o no, si es redondeada o más puntiaguda... la distancia y forma de las vénulas o el color, el algoritmo de clasificación uniría todas esas características para detectar qué tipo de hoja es. Se podría ver como una clasificación más cercana a nuestra interpretación.

Las dos técnicas confluyen en el mismo proceso, siendo necesarios la obtención de ambos. La siguiente Figura 2.3 ilustra un proceso muy simple de extracción de características.

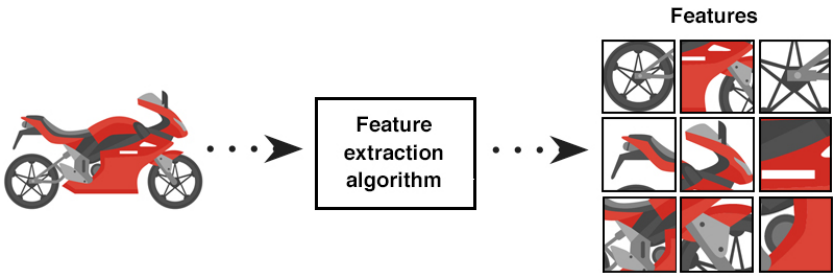


Figura 2.3: Proceso de extracción de características. Adaptado de [2]

Métodos para Precisar los Resultados Obtenidos

Es bastante común asumir que los resultados obtenidos si tienen errores, es consecuencia única de un problema en la clasificación, lo cual es falso, por ello, antes de implementar un sistema para precisar los resultados obtenidos se debe identificar la fuente de los errores [30]. Normalmente se hace una generalización de los resultados obtenidos, dado que para tener la certeza de que son del todo correctos, se debería disponer de datos infinitos.

Los errores se pueden dar, además de la clasificación, por ejemplo, por una fuente de datos de entrenamiento pobre, mala calidad de las imágenes, errores de preprocesamiento de la imagen o mala calidad del preprocesamiento. Es importante detectar la fuente de errores en vez de asumir que es por el algoritmo de clasificación usado y antes de empezar a medir la precisión de los resultados [19].

Uno de los métodos más utilizados para evaluar la precisión de los resultados obtenidos es la matriz de confusión.

Una matriz de confusión de un problema con N clases es una matriz NxN en la que las filas son las clases reales y las columnas las clases predichas por el modelo, ilustrado en la Figura 2.4, con lo que nos dice la precisión con la cual una clase ha sido confundida con otra [31].

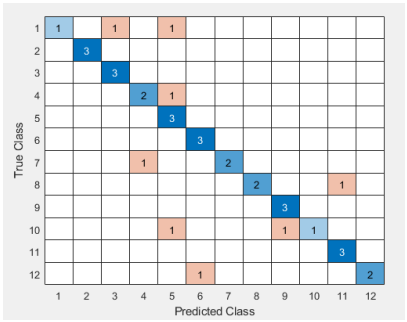


Figura 2.4: Ejemplo de matriz de confusión

2.3. Bases de datos existentes.

La construccion y elección de una base de datos acorde a un proyecto es un paso crítico el cual puede implicar el fracaso del mismo si se toma una mala decision. Para llevar a cabo este proyecto se investigaron diversas fuentes de datos, con el fin de conocer cómo afrontar su escenario y sus peculiaridades.

Portales como Kaggle, Image-Net o Arxiv.org pueden ser un punto de inicio para la búsqueda de datasets e investigaciones, aunque, es enorme la cantidad de datos existentes, con lo cual, si no tenemos bien claras las necesidades y acotamos, se puede convertir en una ardua tarea. Para este proyecto en concreto se buscaron investigaciones y bases de datos relacionadas con la clasificación de especies de hojas.

De entre todas las bases de datos encontradas, se eligieron las siguientes para investigar más en profundidad y ver de qué manera se estaban utilizando, y así sacar conclusiones y realizar el proyecto. Muchos de los encontrados consistían en una clasificación binaria de si era o no una hoja y de si la hoja estaba o no sana, escenarios aplicables a la vida real, pero no el objeto de nuestra investigación, con lo cual quedaron descartados.

2.3.1. Leaf Det

Un dataset binario, encontrado en Kaggle [32], preparado para modelos que hacen decisiones de si la imagen es una hoja o no lo es. Las hojas vienen ya segmentadas por su silueta. Es un buen punto de inicio, dada la segmentación de las hojas se abarata el coste de preprocesarlas y se puede entrenar directamente un modelo para clasificarlas.

Este dataset sirvió para, a la hora de construir el de este proyecto, segmentar las hojas, lo cual es útil cuando además de la hoja hay información en su fondo, eliminándola ya que no es relevante.

2.3.2. LeafSnap Dataset

Es el dataset, encontrado también en Kaggle [33], con el mayor número de especies de los 3 mencionados, contiene 184 especies, clasificadas por su nombre científico, teniendo varias integrantes de la misma familia, proporcionando una buena cantidad de datos. Las imágenes cuentan con fondo liso para hacer más sencillo el trabajo a la red convolucional, o para hacer una segmentación propia. Nos dan también un directorio con una segmentación hecha, cuyo resultado es la silueta y contenido de la hoja en fondo blanco y el resto en fondo negro, pudiéndose hacer una clasificación por silueta [34]. Es otro ejemplo del dataset en el cual se basó este proyecto, y es el más completo encontrado.

2.3.3. Flavia Dataset

Contiene variadas especies, algunas de la misma familia, no tantas como el *LeafSnap Dataset*, pero con un buen número de muestras por hoja. Las imágenes son de buena calidad, tienen una alta resolución, las hojas son a color sobre fondo blanco, con lo cual no haría falta segmentarlas. El dataset no viene con las hojas clasificadas por defecto, es tan solo un directorio con las hojas identificadas por un identificador, por lo que si queremos organizarlas, los autores facilitan un CSV para ello. Los autores de este dataset explican que debido a los problemas que encontraron al no haber un dataset estándar de hojas, decidieron compartir este para el trabajo de futuras investigaciones [35].

2.3.4. Leaf Dataset

Se encontró un dataset con diversas especies existentes en la península ibérica, y debido al alcance del proyecto, se decidió utilizar este. Contiene diversas especies y algunas coincidentes en la familia. Presenta imágenes de buena calidad, con la hoja a color y un fondo rosa [10]. Para el desarrollo de los experimentos se acotó el dataset.

2.4. Estudios realizados

Dada la existencia de tantas especies y tantas familias, guardando similitud algunas con otras, es un campo que resulta muy útil en el campo de la clasificación de imágenes y visión por computador. Diversas investigaciones se han llevado a cabo a lo largo del tiempo, utilizando variados métodos, como pueden ser *Support Vector Machine (SVM)*, *Probabilistic Neural Networks (PNN)*, *General Regression Neural Networks (GRNN)* o *Convolutional Neural Network (CNN).*

2.4.1. Leaf Classification Using Shape, Color, and Texture Features

Las investigaciones que se habían hecho hasta ahora no solían tener en cuenta el color de la hoja, pues pensaban que no sería significativo, pues según la estación y la hoja, el color es cambiante. En esta investigación tuvieron en cuenta la forma, las venas, las texturas y el color de la hoja. Para llevar a cabo la identificación, crearon una Red Neuronal Probabilística (PNN), consiguiendo un acierto medio del 93.75 % [3].

En primera instancia, definieron como iban a realizar la extracción de características, eligiendo diversas características físicas de la hoja, tales como las características de forma, de color, de las venas y de texturas.

Una vez obtuvieron las características definidas, se llevó a cabo una normalización de éstas, dado que estaban en rangos diferentes, y una característica en un rango mayor tendría más relevancia en la función de coste del clasificador. El sistema que presentan se representa en la Figura 2.5.

Primero, se selecciona una imagen y se usa como input del sistema. Se segmenta y se obtienen como resultado una imagen segmentada por contorno, obtenida convirtiendo la original a escala de grises y de ésta a binario (negro - blanco) mediante umbralización adaptativa. Después, se corrigen los agujeros aparecidos en la hoja, causados por el umbral adaptativo. Una vez se obtiene la imagen segmentada, se une con la original para eliminar el fondo y así facilitar la obtención de características de forma [3].

Con la imagen segmentada, se calculan las características de forma. color, textura y forma. Estas características se clasifican en la PNN, que ha sido entrenada previamente. La red neuronal

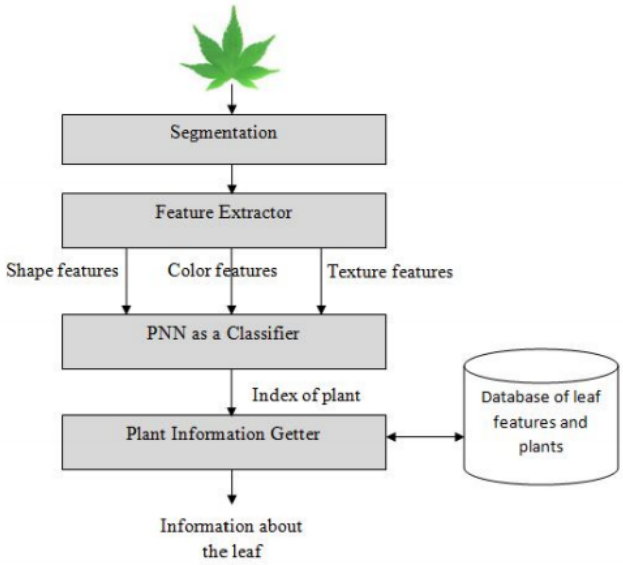


Figura 2.5: Sistema del estudio. Extraído de [3]

produce un índice indicando a qué especie pertenece, la cual se traduce con el componente "Plant Information Getter", que consulta con el índice la especie de la planta [3].

Para probar el sistema propuesto utilizaron la base de datos Flavia. Utilizaron 40 especies de plantas para entrenar el clasificador y 10 plantas por especie para comprobar su rendimiento.

La fórmula 2.1 se utiliza para calcular el rendimiento.

$$Performance = \frac{n_r}{n_t} \tag{2.1}$$

Donde n_r es el número de imágenes relevantes para el caso y n_t es el total introducido [3].

Hicieron varios experimentos, cuyos resultados han sido recogidos en la Figura 2.6.

VARIOUS FEATURES AND ITS PERFORMANCES

Features	Performance
PFT	74.6875%
PFT + 3 geometric features	77.5000%
PFT + 3 geometric features + mean of colors	82.5000%
PFT + 3 geometric features + mean of colors + standard deviation of colors	88.1250%
PFT + 3 geometric features + mean of colors + standard deviation of colors + skewness of colors	88.7500%
PFT + 3 geometric features + mean of colors + standard deviation of colors + skewness of colors + kurtosis of colors	87.8125%
PFT + 3 geometric features + mean of colors + standard deviation of colors + skewness of colors + kurtosis of colors + 12 texture features	90.6250%
PFT + 3 geometric features + mean of colors + standard deviation of colors + skewness of colors + 12 texture features	90.0000%
PFT + 3 geometric features + 12 texture features	85.3125%
PFT + 3 geometric features + mean of colors + standard deviation of colors + skewness of colors + 12 texture features + 3 vein features	93.7500%
PFT + 3 geometric features + mean of colors + standard deviation of colors + skewness of colors + kurtosis of colors + 12 texture features + 3 vein features	93.4375%

Figura 2.6: Resultados del estudio. Extraído de [3]

Viendo los resultados, obtienen como mejor resultado un rendimiento del 93.75 %, usando características de forma, color sin curtosis, venas y textura. Se aprecia que todas las características son significativas excepto la curtosis.

2.4.2. LeafNet: A computer vision system for automatic plant species identification

Con la motivación de facilitar el flujo de trabajo dentro del campo de la investigación vegetal y la identificación de plantas a usuarios inexpertos, construyeron una aplicación móvil para la identificación automática de hojas mediante una fotografía.

Descartan la aproximación del uso completo en la identificación de la *Visión por Computador* por utilizar unas técnicas muy manuales, dependientes de las características elegidas y del modelo creado, los cuales no tienen porqué ser válidos de manera universal, dada la enorme cantidad de especies de plantas existentes [4]. A este enfoque, con el fin de resolver sus limitaciones, se le suma el uso del *Machine Learning*. Las hojas necesitan, en modelos supervisados, ser identificadas y clasificadas para un previo paso de entrenamiento. Es así también, que al desarrollar una aplicación dependiente de algoritmos de Machine Learning, se necesita continuo mantenimiento y mejora del algoritmo, siendo dependiente de la eficacia de los resultados en cuanto que se necesita intervención humana para contrastarlos y mejorarlos.

Un importante paso es el de deshacerse de la necesidad de que las características sean modeladas por un ser humano. Las *Redes Neuronales Convolucionales (CNN)* han supuesto un avance significativo en el campo de la *Visión por Computador*, especialmente en la categorización e identificación pues han permitido una obtención automática de representaciones y características, gracias también a la mayor disponibilidad de mayor cantidad de datos, necesarios para el entrenamiento de este tipo de redes [4].

Las CNN han supuesto un gran avance en la categorización de *grano fino* [4], donde normalmente existe una variación muy sutil entre elementos de intra-clases pero una gran variación entre elementos inter-clases, los cuales sólo eran reconocibles por expertos. La clasificación de hojas es un ejemplo de esto, puesto que hojas de la misma subespecie pueden ser muy diferentes pero hojas de clases diferentes pueden parecerse mucho.

El **dataset** que utilizan para entrenar y hacer tests de su CNN es el *LeafSnap Dataset* que está compuesto de dos subsets, el primero con imágenes de laboratorio, de alta calidad y con una iluminación controlada, y el segundo con imágenes de campo tomadas por usuarios con dispositivos móviles, donde se aprecian variaciones de desenfoque, iluminación, sombras, ruido... pero con un fondo uniforme, como las imágenes de laboratorio.

Para corregir el sobreajuste *(Overfitting)*, se aplican transformaciones a las imágenes como pueden ser girarlas, descentrarlas o cambiarles la escala. Con esto consiguen un set de entrenamiento más grande y no uniforme que ayuda en la clasificación.

Puesto que las CNN necesitan que los datos de entrada tengan todos la misma escala hacen una estandarización desescalando las imágenes a un tamaño de 256x256p. Además, la media de cada canal RGB la restan de los valores RGB de todos los píxeles para estandarizar el dataset también [4].

Para la CNN utilizan un framework llamado *Caffe*, desarrollado por el "Berkeley Vision and Learning Centerz contribuciones de la comunidad. *Caffe* soporta la interfaz de programación *CUDA* mediante *GPUs* de *NVIDIA*, con la librería de Deep Learning *CuDNN* [4].

La **arquitectura** de la CNN se basa en módulos que utilizan la técnica de *Dimension Reduction (Reducción de la Dimensionalidad)*, el cual consiste en la compresión de un alto

número de características en una subcaracterística de menor dimensión sin perder la información importante. Cada módulo tiene 2 capas convolucionales seguidas de una capa de *Agrupación Máxima (MAX-Pooling)* con un filtro 2x2 que reduce a la mitad el ancho y el alto. Las capas *MAX-Pooling* son esenciales al construir CNN, pues reducen de manera dramática el número de conexiones entre la última capa convolucional y la primera capa *Totalmente Conectada (Full Connected Layer)*.

Construyen entonces su arquitectura con 5 módulos como este, sumada a una capa final *MAX-Pooling* con un filtro 2x2, la cual alimenta 3 capas *Fully Connected*. Los primeros 5 módulos aprenden en la fase de entrenamiento y clasifican en la fase de clasificación, mientras que las Full-Connected hacen ambas cosas en ambas fases.

El diagrama de la arquitectura es el siguiente 2.7:

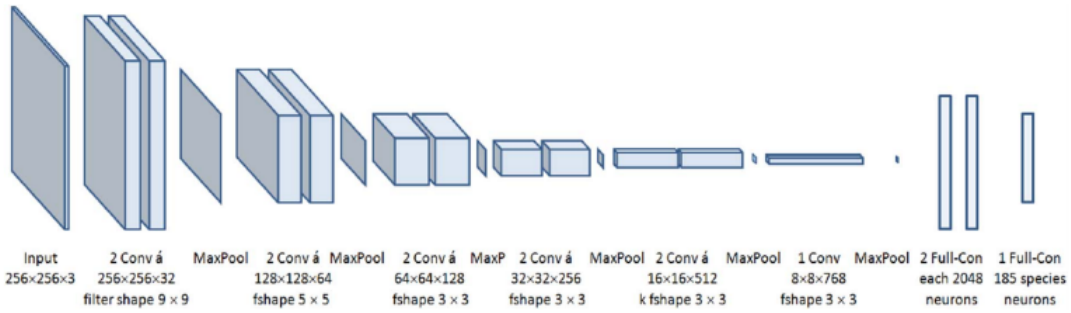


Fig. 3. LeafNet architecture in a simplified visualization (cf. text).

Figura 2.7: Arquitectura de la red. Extraída de [4]

Los filtros utilizados en las primeras capas son de mayor tamaño para capturar regiones más grandes del raster de entrada, optimizando la convolución. La reducción del filtro y el crecimiento del número de mapeos de características en las sucesivas capas proporciona una mayor diversidad y complejidad de características. Lo ilustran en la Figura 2.8.

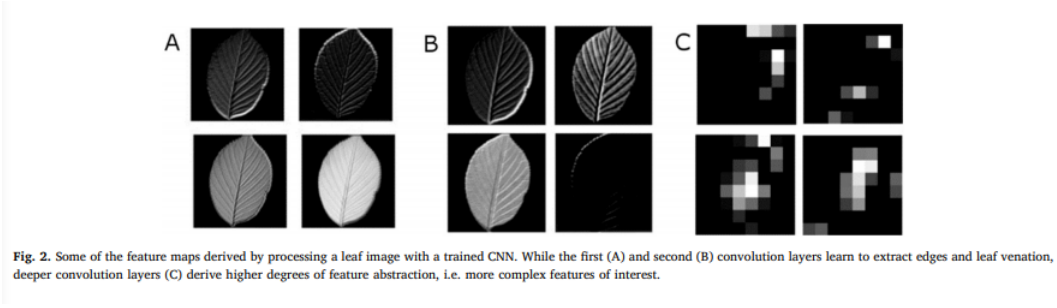


Fig. 2. Some of the feature maps derived by processing a leaf image with a trained CNN. While the first (A) and second (B) convolution layers learn to extract edges and leaf venation, deeper convolution layers (C) derive higher degrees of feature abstraction, i.e. more complex features of interest.

Figura 2.8: Extracción de características. Extraída de [4]

Entrenaron su red con el dataset *LeafSnap* y con otros dos datasets públicos, *Foliage* y *Flavia*. Para cada dataset adecuaron las características de la red.

Los resultados que obtuvieron para el dataset *LeafSnap*, utilizando únicamente imágenes con información de fondo (pues se acerca más al propósito de la aplicación), es de un 86.3 % acertando exactamente la clase y un 97.8 % para un top de 5 clases. La matriz de confusión es la representada en la Figura 2.9.

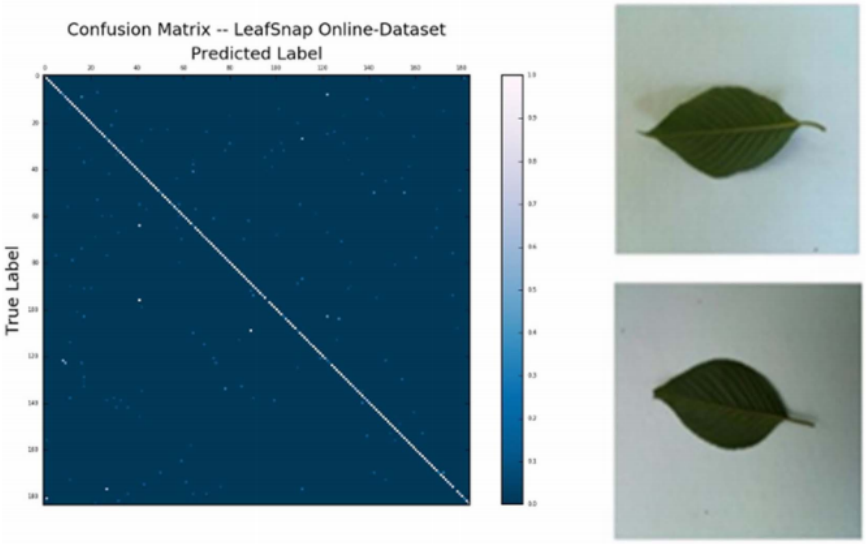


Fig. 4. Left: The confusion matrix for the LeafSnap online dataset. Colours indicate the proportion of leaf images from the actual species that our LeafNet CNN classified into the predicted categories. Right: leaf images of the species *Prunus subhirtella* (top) and *Prunus virginiana* (bottom) (Kumar et al., 2012). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Figura 2.9: Matriz de confusión. Extraída de [4]

La diagonal de la matriz de confusión representa las clases clasificadas adecuadamente, apreciándose que la mayoría de las clases están clasificadas correctamente. Explican que las clases lejanas a la diagonal pueda ser por gran similitud en la imagen de entrada con clases incorrectas o por especies de las que disponían muy pocos datos de entrenamiento [4]. Los resultados obtenidos con los datasets *Foliage* y *Flavia* son los siguientes. El dataset *Foliage* contiene 60 especies, cada una con 120 imágenes. Consiguen un 95.8 % acertando exactamente y un 99.6 % con un top 5. El dataset *Flavia* consiste en 32 especies con un número de imágenes por especies de 50 a 60. Consiguen un 97.9 % clasificando exactamente y un 99.9 % con un top 5.

Las matrices de confusión 2.10 son las siguientes.

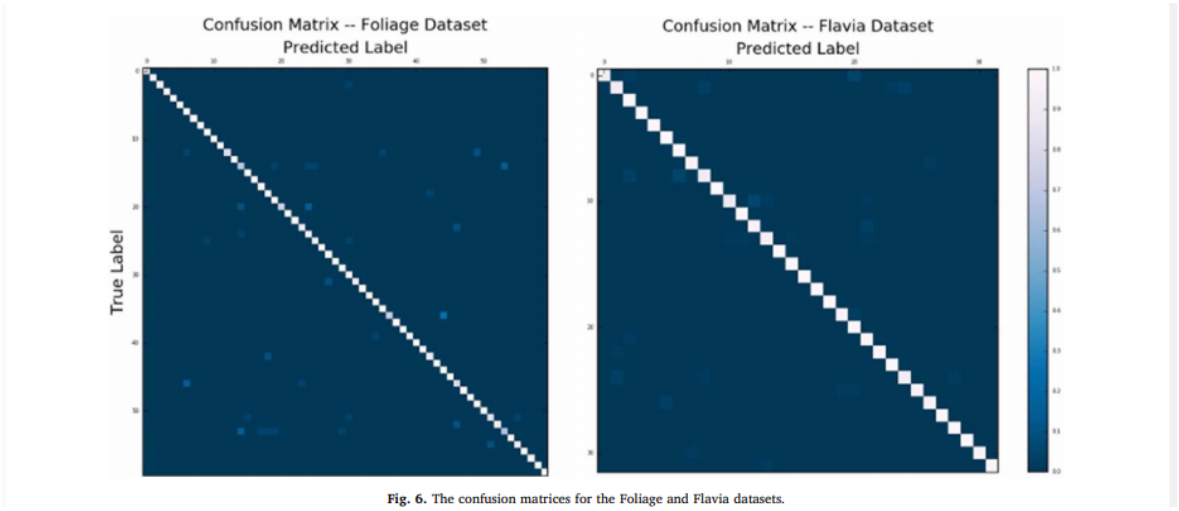


Fig. 6. The confusion matrices for the Foliage and Flavia datasets.

Figura 2.10: Matrices de confusión usando los datasets Foliage y Flavia. Extraído de [4]

Las dos matrices de confusión muestran diagonales claras, confirmando los resultados obtenidos.

Los resultados de la clasificación los obtuvieron siguiendo diferentes métricas de evaluación como son el *Average Top-N Accuracies*, *Average Scores*, *Mean Average Precision* (MAP) [4].

La Figura 2.11 ilustra los resultados que obtuvieron siguiendo estas tres métricas para los datasets Flavia, Foliage Y LeafSnap.

Table 2
Results on the Flavia dataset, Foliage dataset and LeafSnap-online dataset.

Dataset	No. of species	Top-1 accuracy	Top-5 accuracy	MRR score	MAP score
LeafSnap	184	86.3%	97.8%	92.2%	83.7%
Foliage	60	95.8%	99.6%	97.6%	95.3%
Flavia	32	97.9%	99.9%	98.8%	97.2%

Figura 2.11: Resultados del estudio. Extraído de [4]

Los resultados que obtuvieron mediante la aproximación que decidieron utilizar fueron prometedores, demostraron que utilizando una red CNN para el aprendizaje de características y clasificación puede tener mejores resultados que aquellos obtenidos mediante la utilización de técnicas de obtención de características diseñadas (*handcrafted features*), como puede ser por ejemplo en este caso, el cálculo de la circunferencia de la hoja o la obtención manual de sus venas [4]. Los resultados de los estudios propios y el estudio que hizo el equipo de LeafNet sobre los datasets *LeafSnap*, *Foliage* y *Flavia* se pueden ver en la Figura 2.12, donde los resultados del Top-1 obtenidos con una CNN son superiores.

Table 3
Top-1 accuracies achieved on the Flavia, Foliage and LeafSnap-online datasets.

Dataset	No. of species	Top-1 acc. of our approach	Top-1 acc. of Wu et al., 2007	Top-1 acc. of Kadir, 2014	Top-1 acc. of Kumar et al., 2012
LeafSnap	184	86.3%	/	/	73.0%
Foliage	60	95.8%	/	95.0%	/
Flavia	32	97.9%	90.3%	97.2%	/

Figura 2.12: Resultados Top-1 del estudio. Extraído de [4]

Dado que las CNN buscan características locales en una pequeña porción de la imagen, pueden entrar al detalle de una manera efectiva y menos trabajosa.

2.5. Segmentación y Detección de Bordes.

La ***Detección de Bordes*** es una aplicación de una técnica del procesamiento de imágenes llamada ***Segmentación*** la cual consiste en la extracción o división de objetos o regiones de una imagen digital. Se puede utilizar para encontrar los límites de los objetos de una imagen o para localizarlos. Lo hace asignando una etiqueta a cada píxel de la imagen, de esta manera, los píxeles que compartan etiqueta tendrán características similares como pueden ser color, textura o intensidad [36].

Esto genera un conjunto de segmentos que cubren la imagen o un conjunto de curvas de nivel. Mientras que los píxeles de una misma región tendrán valores similares por tener características

similares, se apreciará gran diferencia con las regiones adyacentes, debido a que sus características serán diferentes. Los algoritmos de segmentación se basan en las *Propiedades de Discontinuidad* o *Similitud de los Píxeles Vecinos* [36].

2.5.1. Detección de Discontinuidades

Los ***Algoritmos de Discontinuidad*** dividen la imagen basándose en los cambios bruscos en los píxeles vecinos para realizar detección de puntos aislados, detección de líneas o detección de bordes [5]. Encontramos tres tipos:

Puntos Aislados:

Un punto aislado en una imagen es aquel cuyo tono de gris difiere de los tonos de gris de sus píxeles vecinos [5]. Para encontrar un punto aislado se aplica una máscara Laplaciana, cuyo resultado es una matriz con un píxel cuyo valor difiere del resto como se aprecia en la siguiente Figura 2.13.

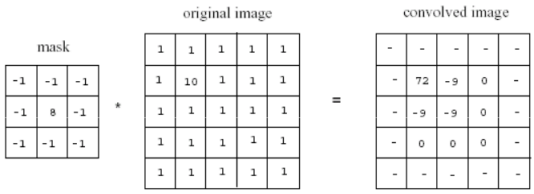


Figura 2.13: Convolución para la obtención de puntos aislados. Extraída de [5]

Líneas:

Son consecutivos píxeles cuyo tono es el mismo pero difieren de los píxeles vecinos. Análogamente se utiliza una máscara Laplaciana, la cual resulta en píxeles con valores distintos a los vecinos, pero iguales entre ellos, donde se aprecia también la dirección que toman los píxeles [5].

Por ejemplo, tendríamos líneas en horizontal, en diagonal y vertical 2.14.

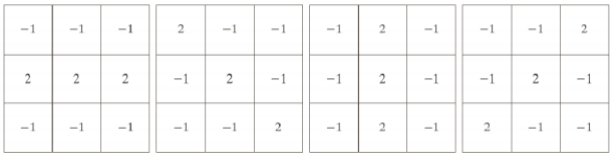


Figura 2.14: Representación de líneas en rásters. Extraído de [5]

Bordes:

Idealmente el cambio entre los píxeles vecinos tiene que ser brusco, como si de un contorno se tratase, pero en la realidad existe una variación entre el borde y los píxeles vecinos. Se resuelve aplicando una derivada sobre los puntos, donde si su resultado es mayor que cierto umbral, se dice que ese punto pertenece al borde. Esto es debido a que existe una gran variación en aquellos puntos cuyo resultado es menor al umbral. Con la segunda derivada podemos saber en qué parte del borde se encuentra el píxel [5].

En cuanto a la detección de bordes se debe tener en cuenta que si hay ruido en la imagen, se detectará como borde. Por ello, se debe primero suavizar la imagen para reducir el ruido para posteriormente poder realizar la detección de bordes. También puede ocurrir que debido a las múltiples texturas de la imagen obtengamos una detección de bordes con *ruido* debido a las texturas. Es por ello que es conveniente realizar una umbralización de la imagen obtenida [5].

En la detección de bordes se debe tener en cuenta que existen falsos positivos (píxeles que en realidad no pertenecen al borde aun siendo detectados como que si) y falsos negativos (píxeles cuya pertenencia a un borde ha sido detectada como negativa cuando es positiva). Es por ello que, o bien se puede comparar el resultado con la imagen original, o mediante un evaluador realizar una comprobación de los píxeles vecinos con el detectado.

2.5.2. Algoritmos de Detección de Similitudes

Los ***Algoritmos de Detección de Similitudes*** buscan similitudes entre las zonas de una imagen para dividirlas según unos criterios prefijados. Nos encontramos con dos tipos, la *División y Fusión* y la *Umbralización*.

División y Fusión

Las imágenes se pueden dividir en subregiones según unos criterios prefijados, de manera que, iterativamente, vamos dividiendo dichas subregiones o fusionándolas hasta que todas ellas cumplan ciertas condiciones.

Umbralización

Sirve para diferenciar un objeto del fondo de la imagen mediante binarización. Se genera un histograma mediante la imagen en nivel de grises y partir del histograma se define un valor umbral, y los valores por debajo de este umbral se transforman a negro y los mayores a blanco.

2.6. Redes Neuronales Convolucionales para Extracción de Características.

Las *Redes Neuronales Convolucionales* o CNNs es un tipo especial de *Redes Neuronales* que intentan emular la conectividad de las neuronas del cortex visual de un cerebro biológico. Su utilidad recae en el campo de la *Visión Artificial*, con tareas como son la detección de bordes o la clasificación y detección de objetos en imágenes o vídeos, entre otras. Esto se consigue emulando el cerebro biológico y su jerarquía, formando capas, donde las primeras detectan, por ejemplo, líneas y curvas, y a medida que se avanza, se van especializando hasta ser capaces de reconocer un animal, por ejemplo.

2.6.1. Arquitectura y Funcionamiento.

Las *Redes Convolucionales* se organizan en bloques de capas llamadas *Capas Convolucionales* a las que después se les añade una función para realizar un mapeo no lineal. Un ejemplo de arquitectura sería el mostrado en la Figura 2.15.

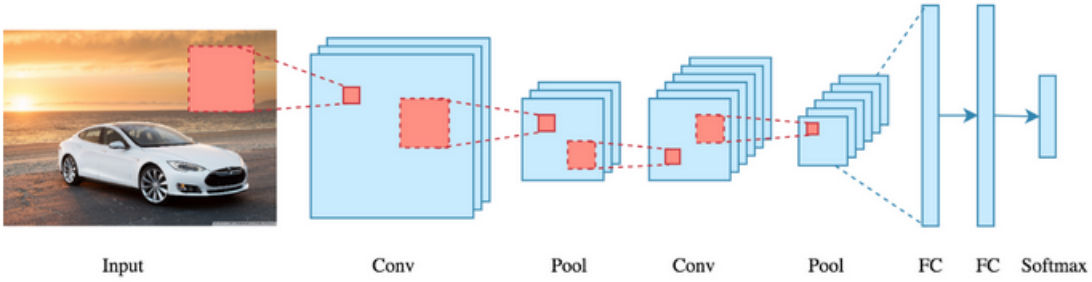


Figura 2.15: Diagrama de una CNN. Extraído de [6]

La red contiene primero una fase de *Extracción de Características*, que está compuesta por *Neuronas Convolucionales* y *Reducción de Muestreo*, las cuales se van alternando para conseguir extraer diferentes características y reduciendo la dimensionalidad, para tener extraidas las características determinantes y que así las neuronas de las capas lejanas se activen con ellas. Las capas más cercanas al input se encargan de conseguir las características más básicas como pueden ser los bordes mientras que las capas más profundas se encargan de tareas de extracción más complejas y específicas como extraer las características de subregiones [4]. Esta especialización es auto-organizada durante el entrenamiento, es decir, no es fruto del diseño.

La fase final se encarga de hacer una clasificación sobre la extracción de características hecha.

La ***Convolución*** es una operación que consiste en filtrar una imagen usando una máscara o kernel, que combinándolas, producen distintos resultados. En la convolución, los píxeles de entrada se combinan mediante cierta función generando un píxel de salida, como se ve en la Figura 2.16. Las *Máscaras o Kernels* representan la conectividad entre las capas sucesivas [37].

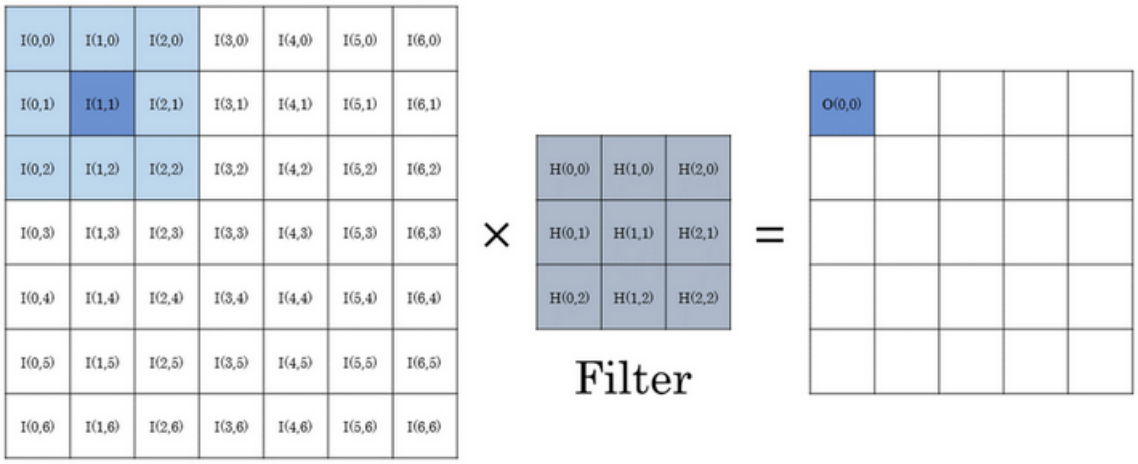


Figura 2.16: Operación de convolución. Extraída de [7]

Las ***Máscaras*** son una pequeña matriz sobre la cual, al convolucionar, se opera toda la imagen de entrada, que se interpreta como una matriz, multiplicando la máscara sobre todas las neuronas de entrada, por orden, generando una nueva de matriz de salida que es la nueva

capa de neuronas ocultas. Conseguimos reducir la imagen de entrada y obtener propiedades relevantes, dadas por el tipo de máscara que se haya aplicado, y descartado otras. No se tiene una sola máscara pues perderíamos mucha información, si no que se tienen varias las cuales forman el llamado filtro. Con este filtro conseguimos no perder tanta información, pues cada máscara habrá conseguido unas propiedades u otras [38]. Con esto, conseguimos tantas matrices de salida como máscaras tenga el filtro.

La reducción de la matriz generada por las máscaras se consigue mediante las llamadas *Capas de Pooling* o *Capas de Reducción*, y la operación recibe el nombre de ***Reducción de Muestreo*** [39]. Reciben como entrada cada una de las matrices de características generadas por las máscaras y obtienen una nueva matriz de características simplificada. La matriz resultado, al estar simplificada, tiene menos información, pero esto no implica que se hayan perdido características relevantes, si no que trae beneficios de optimización, ya que las siguientes capas no tendrán sobrecarga de cálculo por la reducción del dimensionamiento de la matriz y el *sobreajuste* u *overfitting* se reduciría [39].

El ***Overfitting*** se produce cuando un modelo funciona bien con los datos de entrenamiento pero no lo hace con los datos de evaluación. Esto se produce por haber entrenado la red con muestras atípicas, con ruido o poco representativas, insuficientes para poder hacer una generalización en la clasificación. Es decir, en evaluación, las muestras se salen del rango establecido en el entrenamiento [39].

La operación que se realiza en la *Reducción de Muestreo* consiste en subdividir la matriz en un conjunto de rectángulos, y sobre ellos, realizar la operación según el tipo de *Capa de Pooling* que se elija [39]. Normalmente las *Capas de Pooling* se clasifican en dos tipos:

- **Maximum Pooling o Max-Pooling**: recoge el valor máximo de los valores pertenecientes al área receptiva.

- **Average Pooling**: calcula el valor medio de los valores pertenecientes al área receptiva.

Los dos tipos de *Pooling* se recogen en la siguiente Figura 2.17.

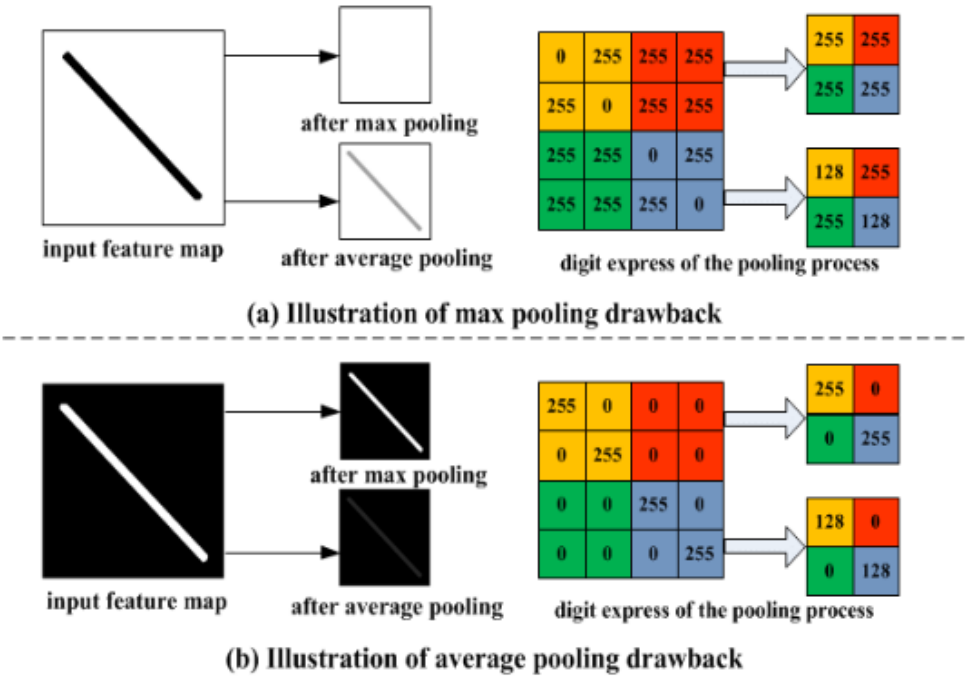


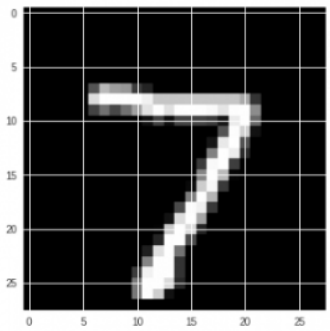
Figura 2.17: Tipos de pooling. Extraída de [8]

Al finalizar los bloques de *Convolución* se encuentran generalmente una red de capas llamadas ***Capas Clasificadoras Totalmente Conectadas***. Se encargan de clasificar las características que portan las matrices, extraídas y depuradas en la *convolución* y *pooling* según los objetivos de entrenamiento. Estas capas se encuentran conectadas a cada píxel de la matriz, siendo cada conexión una neurona independiente y funcionan como las de un *Perceptrón Multicapa*, donde la salida de cada neurona se calcula multiplicando la salida de la capa anterior por el peso de la conexión y aplicando a ese resultado una función de activación. La última capa de esta red de capas tendrá tantas neuronas como clases deba predecir [39].

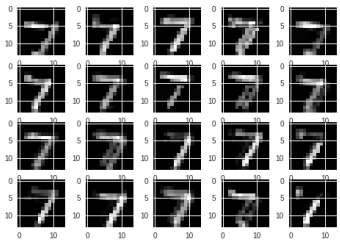
2.6.2. Entrenamiento de CNNs

El entrenamiento de las *CNNs* es similar al de las *Redes Neuronales* convencionales salvo por el detalle de que se deben ajustar los pesos de los distintos *kernels* de las distintas *Capas Convolucionales*.

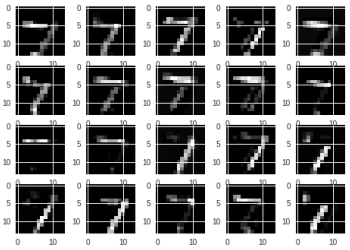
El proceso por el cual la información se recoge en las distintas capas es el mencionado antes, las primeras capas retienen mucha información y a medida que va recorriendo la red, la información retenida se convierte en tan sólo las características más relevantes, lo cual permite a la red realizar una generalización. En las siguientes Figuras se ilustra la información retenida por las primeras capas 2.18b y por las avanzadas 2.18c.



(a) Imagen de entrada para visualización de características. Extraída de [40]



(b) Visualización de la primera convolución. Extraída de [40]



(c) Visualización de la última convolución. Extraída de [40]

Figura 2.18: Visualización de las capas

A medida que la imagen avanza en la red, se aprecia que las características capturadas por las convoluciones van de muy generales a más específicas [40].

2.6.3. Backpropagation y Feed-Forward.

Existen dos topologías de *Redes Neuronales* para entrenamiento, son las llamadas ***Feed-Forward o Prealimentada*** y Back-Propagation o Propagación hacia atrás, las cuales sirven para alimentar las capas de la red con la información de entrada y generar unos datos de entrenamiento.

El **Feed-Forward** es la topología de red más habitual en la práctica. Se basa en que las conexiones de las neuronas no forman un ciclo, la entrada se va propagando por la red hacia delante, sin retroalimentar a neuronas anteriores tal como muestra la Figura 2.19. No existe autoadaptación de pesos en las neuronas, se basa más en el *ensayo y error* [9].

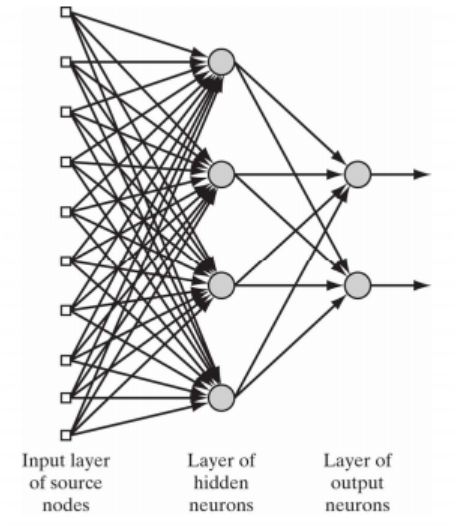


Figura 2.19: Topología feed-forward. Extraída de [9]

El **Backpropagation** es una técnica de programación para hacer una búsqueda sistemática a través de todas las configuraciones posibles dentro de un espacio de búsqueda empleando un ciclo de propagación del resultado obtenido [9].

La red se alimenta, se propaga por las capas y se genera una salida la cual se compara con la salida deseada y se calcula una señal de error para cada una de las salidas. Una fracción de la salida de error se propaga hacia atrás partiendo de la capa de salida hacia todas las neuronas de la capa que contribuye a la de salida directamente. La fracción de la salida de error es la contribución relativa que haya aportado cada neurona a la salida original. Este proceso se repite recorriendo todas las capas hasta que todas las neuronas de la red hayan recibido una señal de error que describa su contribución relativa al error total [9].

Con este proceso, a medida que la red se va entrenando, las neuronas de las capas intermedias se organizan a sí mismas de manera que las distinas neuronas aprender a reconocer las distintas características de las totales de la entrada.

2.6.4. Requisitos software. MATLAB.

MATLAB proviene del acrónimo en inglés *MATrix LABoratory*. Es un software propietario matemático optimizado para resolver problemas científicos y de ingeniería a través de su lenguaje *M*. El software cuenta con un IDE y una amplia y activa comunidad donde es posible utilizar herramientas e implementaciones creadas por ella, los cuales se denominan *Toolboxes*.

MATLAB en sus inicios fue escrito a finales de la década de los 70 en FORTRAN por Cleve Moler e integrantes de los proyectos *LINPACK* y *EISPACK* lo ampliaron desarrollando los algoritmos matriciales. Actualmente *MATLAB* está escrito en C por la empresa propietaria *Mathworks*. En 1984 fue lanzado al mercado, teniendo buena acogida entre la comunidad científica y universitaria, aunque al ser software propietario tuvo también bastantes críticas por parte de la comunidad del software libre, ya que los usuarios están sujetos a la empresa propietaria *Mathworks* [41].

Las aplicaciones desarrolladas con *MATLAB* son escritas en *M*, un lenguaje de programación multiparadigma propio, el cual es interpretado y débilmente tipado. Sus funciones principales consisten en la manipulación de matrices, la implementación de algoritmos, la representación gráfica de funciones y datos, la creación de GUIs y la comunicación con hardware y otros lenguajes de programación como pueden ser C o FORTRAN mediante *wrappers*. Soporta también la programación orientada a objetos y el cálculo lambda [41].

Los *Toolboxes* son archivos empaquetados creados por la comunidad de *MATLAB*, y creados también por la empresa, que incluyen código, apps, datos y documentación con implementaciones ya hechas, y que se pueden instalar y utilizar fácilmente en el entorno. Sería el equivalente a una biblioteca en C. Los *Toolboxes* son muy diversos, encontramos implementaciones, por ejemplo, para computación paralela, análisis de texto, deep learning, machine learning o tratamiento de imágenes.

2.7. Clasificación y Regresión.

Para la evaluación de los resultados de una *Red Convolucional* se pueden utilizar métodos como los mencionados en la sección de *Estudios Realizados*, pero hay multitud de maneras. Una manera es siguiendo mediante la utilización de técnicas de *Clasificación y Regresión*, los cuales se encuentran dentro de los campos del Aprendizaje Automático y la Estadística.

La ***Clasificación*** consiste en la identificación de a qué conjunto de categorías o clase pertenece una muestra, dado un conjunto de muestras sobre el cual se hace cierta recopilación de información y sobre las cuales se conoce la clase a la que pertenecen. Es un problema dentro del *Aprendizaje Automático* que se encuentra dentro de la categoría de *Aprendizaje Supervisado*, en el cual se dispone de antemano las clases a las que pertenecen un grupo de muestras.

La ***Regresión*** es un proceso dentro del campo de la estadística el cual busca estimaciones de las relaciones entre variables, una dependiente (Y) entre una o más independientes ($X_1, X_2, \ldots, X_n$). Para estudiar la relación entre ellas se tiene que analizar la gráfica que proyectan. A esta gráfica se le llama *Gráfica de Dispersión*, y nos permite ver si existe relación entre las variables o no. Existen varias relaciones entre las variables, según el tipo de relación puede ser lineal o no lineal y según el sentido de la relación entre las variables puede ser directa o inversa [42].

3

Diseño y desarrollo

3.1. Esquema General del Algoritmo Propuesto.

En este capítulo se ilustrará la arquitectura del algoritmo propuesto y sus implementaciones, del origen de los datos, el proceso de segmentación, caracterización, comparación y agrupación.

Arquitectura: se ha elegido una arquitectura clásica en cuanto a la clasificación de imágenes con *Deep Learning*. Se ilustra en la Figura 3.1.

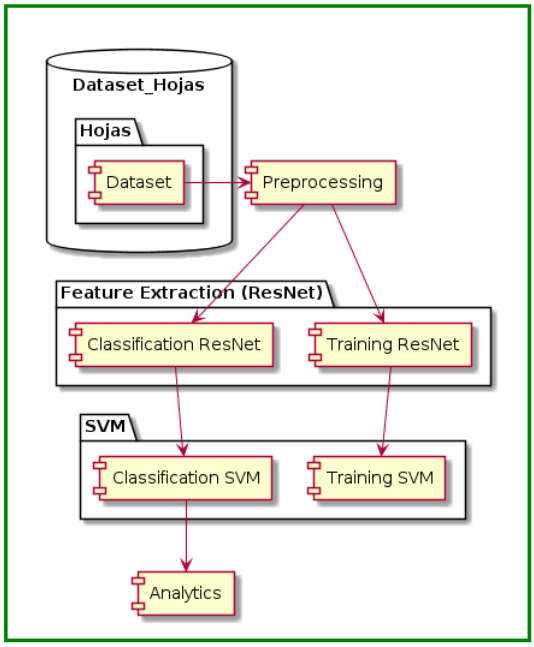


Figura 3.1: Arquitectura del Sistema

Preprocesamiento: se realizaron pruebas con y sin segmentación.

Para las pruebas realizadas con segmentación, se somete al dataset a una extracción del objeto a identificar mediante un filtro de color con el rango RGB del fondo de la imagen, y

una vez extraído, la imagen se inserta sobre un fondo blanco. El siguiente paso llevado a cabo es el de estandarizar el tamaño de todas las imágenes del dataset. En las pruebas realizadas sin segmentación, el dataset tan sólo se somete a un proceso de estandarización del tamaño del dataset. Se realiza también una conversión de escala de grises a RGB, por si hubiera imágenes en dicha escala, pues la red no admite imágenes en escala de grises.

Extracción de Características: una vez se ha realizado el preprocesamiento, se divide el dataset en datos de entrenamiento y datos de test, particionando el 40 % de los datos para entrenamiento y el 60 % para test. La división escoge muestras aleatorias del conjunto de datos.

El conjunto de entrenamiento se introduce en la red, una *ResNet-50*, la cual se encuentra ya preentrenada, que actúa como un extractor de las características de nuestro dataset. Una vez se han extraído las características, repetimos el proceso para el conjunto de datos de test en la red. Con esto conseguimos las características de entrenamiento y test para clasificarlas usando una SVM.

Por cada proceso de extracción y características se evalúa una capa distinta de la red, con el fin de medir la eficacia según la profundidad de la red, ya que las primeras capas se orientan a características más básicas y las más profundas a características más complejas.

Clasificación: introducimos las características extraídas del conjunto de datos de entrenamiento para entrenar la SVM, con la que una vez ha sido entrenada, clasificamos nuestras características del conjunto de datos de test. Una vez ha clasificado, construimos una matriz de confusión de la cual calculamos la media de aciertos. Este proceso se repite también según la capa de la red que estemos evaluando.

3.2. Dataset

El origen del dataset utilizado es el llamado *Leaf Dataset* [10], compuesto por 40 especies diferentes de plantas. La Figura 3.2 ilustra cuales.

Class	Scientific Name	#	Class	Scientific Name	#
1	Quercus suber	12	21	Fraxinus sp.	10
2	Salix atrocinera	10	22	Primula vulgaris	12
3	Populus nigra	10	23	Erodium sp.	11
4	Alnus sp.	8	24	Bougainvillea sp.	13
5	Quercus robur	12	25	Arisarum vulgare	9
6	Crataegus monogyna	8	26	Euonymus japonicus	12
7	Ilex aquifolium	10	27	Ilex perado ssp. azorica	11
8	Nerium oleander	11	28	Magnolia soulangeana	12
9	Betula pubescens	14	29	Buxus sempervirens	12
10	Tilia tomentosa	13	30	Urtica dioica	12
11	Acer palmatum	16	31	Podocarpus sp.	11
12	Celtis sp.	12	32	Acca sellowiana	11
13	Corylus avellana	13	33	Hydrangea sp.	11
14	Castanea sativa	12	34	Pseudosasa japonica	11
15	Populus alba	10	35	Magnolia grandiflora	11
16	Acer negundo	10	36	Geranium sp.	10
17	Taxus bacatta	5	37	Aesculus californica	10
18	Papaver sp.	12	38	Chelidonium majus	10
19	Polypolium vulgare	13	39	Schinus terebinthifolius	10
20	Pinus sp.	12	40	Fragaria vesca	11

Figura 3.2: Dataset de Leaf Dataset. Extraído de [10]

Cada especie es una clase y el símbolo # representa el número de muestras por clase.

Cada especímen fue fotografiado por los autores sobre un fondo coloreado y uniforme, siempre el mismo, utilizando como dispositivo de entrada un *Apple IPAD 2*. Las imágenes son de 24bit RGB y cuentan con una resolución de 720x920 píxeles [10].

De este dataset seleccionamos 12 especies de hojas. En la Tabla 3.1 se ilustran las especies elegidas, su clase y su número de muestras.

Clase	Especie	#
1	Quercus Suber	12
2	Salix Atrocinerea	10
3	Populus Nigra	10
4	Crataegus Monogyna	8
5	Nerium Oleander	11
6	Celtis SP	12
7	Corylus Avellana	13
8	Castanea Sativa	12
9	Populus Alba	10
10	Taxus Bacatta	5
11	Pinus SP	12
12	Fraxinus SP	10

Cuadro 3.1: Tabla con las clases elegidas del dataset de Leaf Dataset

3.3. Pre-procesamiento de la Hoja

Para una de las pruebas se realizó segmentación en las imágenes.

Se utilizó un filtro de color mediante el cual se introduce un rango de los canales RGB el cual contiene los colores del fondo de la imagen, que en este caso es un rango de rosas, debido a la iluminación y a la sombra que arroja la imagen sobre el fondo. El procedimiento es el siguiente:

Para comenzar, se debe cargar la imagen y se define un umbral para generar una máscara, que en este caso corresponde a el fondo de las imágenes que es un rango de color rosa. Se genera entonces la máscara aplicando dicho rango sobre la imagen. Se compone otra imagen con la máscara y la imagen original, filtrando la imagen original con la máscara, con el propósito de conseguir un fondo negro con el objeto segmentado. Finalmente, se genera una última imagen a partir de esta última, convirtiendo el fondo de color negro en color blanco. Este procedimiento se ilustra en la Figura 3.3:

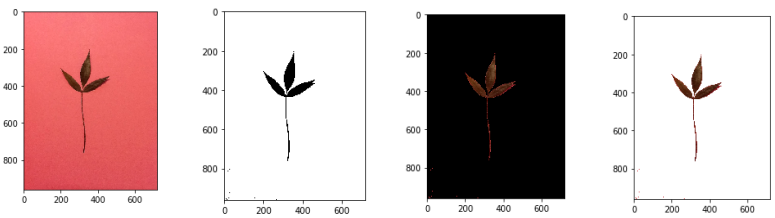


Figura 3.3: Proceso de segmentación

Con este proceso, conseguimos segmentar nuestro dataset y tener un fondo blanco plano, con el fin de eliminar el fondo de color rosa con diferentes tonos por sombras e iluminación.

3.4. Extracción de Características.

Para la extracción de características sobre nuestras imágenes hacemos uso de una *ResNet-50*, una Red Neuronal Residual compuesta por 50 capas. Esta red se encuentra preentrenada con más de un millón de imágenes de la base de datos de *ImageNET*, siendo capaz de clasificar alrededor de 1000 clases, ninguna de ellas entre el conjunto propuesto.

3.4.1. Redes Neuronales Residuales.

Las ***Redes Neuronales Residuales*** o ***Residual Neural Network (ResNet)*** se basan en cierto funcionamiento de las neuronas del cerebro humano, las cuales en sus conexiones realizan saltos entre neuronas sin necesidad de pasar por neuronas intermedias.

Estudios realizados comprobaban la importancia de que para hacer una más potente clasificación era necesario aumentar la profundidad y número de capas de las *Redes Neuronales Convolucionales Profundas*, aunque no es necesariamente cierto que a mayor número de capas se consigue una mejor clasificación.

A medida que se va recorriendo la red, a mayor profundidad la precisión en el entrenamiento comienza a converger hasta que llega a un punto de saturación y se degrada. Esta degradación quiere decir que no todos los sistemas se optimizan de la misma manera, que no basta con simplemente añadir capas a la red. De esta forma, tampoco es posible mejorar los resultados añadiendo las mismas capas varias veces a una red, los resultados obtenidos no mejoran [43].

Las ***Redes Neuronales Residuales*** introducen el funcionamiento de las conexiones neuronales mencionado, utilizando conexiones que saltan entre las capas (*shortcut connections*) convolucionales, existiendo también conexiones directas entre las capas. Suponen que si las capas conectadas se pueden ajustar a una función, las capas que saltan a otras capas convolucionales pueden hacerlo a una función residual [43]. La Figura muestra la arquitectura de una ResNet 3.4.

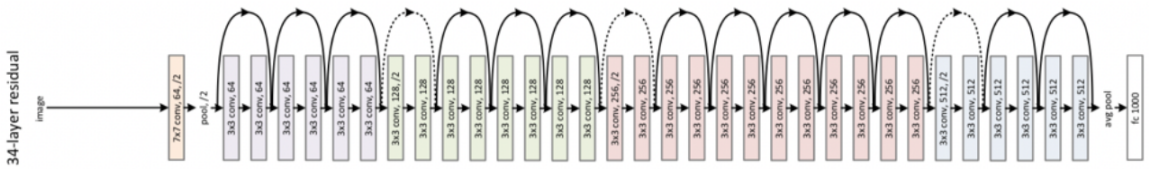


Figura 3.4: Arquitectura de una ResNet. Extraída de [11]

El resultado de las *shortcut connections* viene dado por el resultado de alguna convolución anterior, y para la siguiente iteración, basta con añadirlo al resultado de la conexión con la que esté conectada, teniendo en cuenta que tienen que ser del mismo tamaño.

Estas redes consiguen ser mejores que sus equivalentes con muchas capas pero sin salto entre capas, consiguiendo mejores resultados en entrenamiento al no perder precisión a medida que se va avanzando en la red, dado que como mínimo, la convolución va a tener el resultado que arrojan las *shortcut connections*. Son útiles en trabajos de reconocimiento de imágenes, donde las Redes Neuronales Convolucionales suelen tener un gran número de capas [43].

Una CNN preentrenada nos asegura que las capas menos profundas de la red son capaces de detectar características básicas, actuando como *capas fijas*, aunque sigue siendo necesario entrenar las capas más profundas, que detectan características más complejas, para hacer la red

afín al problema planteado. Es por ello que por lo general, las capas más complejas son aquellas a las que corresponde detectar las características específicas de las imágenes.

Una vez las imágenes han sido preprocesadas, se dividen en dos conjuntos, un 40 % de las imágenes para la extracción de características de entrenamiento y un 60 % para la extracción de características de test. En este punto es donde se realiza la conversión de las imágenes en escala de grises a RGB, por si las hubiera.

Extraemos las características del conjunto de entrenamiento y del conjunto de test usando como parámetros la capa elegida de la red y un *Mini-Batch* de 16 imágenes y obtenemos también las clases originales del conjunto de entrenamiento.

3.5. Clasificación.

La clasificación de las características obtenidas durante el proceso de extracción de características la realizamos por medio de una **SVM**. Como alternativa, podríamos haber reentrenado la última capa de la red, la cual corresponde con el clasificador. En este trabajo buscamos evaluar el rendimiento al combinar un modelo de caracterización con uno diferente de clasificación.

3.5.1. Support Vector Machine

Las ***Support Vector Machine*** o ***SVM*** es un algoritmo de *Aprendizaje Supervisado* dentro del subconjunto de problemas de *Clasificación* y *Regresión*. Su base matemática se encuentra dentro del concepto del *Hiperplano*.

Un *Hiperplano*, dado un espacio con p dimensiones, es un subespacio plano y afín cuyas dimensiones son $p-1$. Por ejemplo, si el espacio es de 3 dimensiones, el hiperplano es de 2. Dadas p dimensiones y los coeficientes del hiperplano $\beta_0, \beta_1, \beta_2, \ldots, \beta_p$, su definición matemática es la siguiente [44]:

$$\beta_0 x_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p = 0 \tag{3.1}$$

Los puntos dados por el vector $x = (x_1, x_2, \ldots, x_p)$ que cumplen la igualdad son puntos pertenecientes al *Hiperplano* [45].

En el caso que x no cumpla la igualdad, estará a un lado o al otro del hiperplano, dándose dos opciones según el signo de la ecuación [44]:

$$\beta_0 x_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p < 0 \tag{3.2}$$

o

$$\beta_0 x_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p > 0 \tag{3.3}$$

El hiperplano divide un espacio de p dimensiones en dos subespacios.

En un principio el algoritmo ***SVM*** se propuso para realizar una clasificación binaria, entre dos clases, pero posteriores investigaciones comprobaron que resultaba posible realizar una clasificación multiclase con él [45].

- Clasificación Binaria.

Supongamos una matriz de dimensiones nxp siendo n el número de muestras y p el número de predictores, donde en la variable resultado existen dos clases distintas dadas $y_1, y_2, \ldots, y_n \in (-1, 1)$, siendo las clases -1 o 1. Se debe resolver la ecuación 3.1 y comprobar el signo del resultado como se ve en las ecuaciones 3.2 y 3.3 [44]:

$$f = \begin{cases} \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} < 0 & y_i = -1 \\ \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} > 0 & y_i = 1 \end{cases} \tag{3.4}$$

para $i = 1, 2, \ldots, n$

Al llegar una nueva muestra a clasificar, para averiguar la clase a la que pertenece, se debe operar la función y en función de si el resultado es negativo o positivo, corresponderá a una clase u otra.

El resultado nos puede servir para medir también la fiabilidad de la clasificación, de forma que, cuanto más se aleje el resultado de 0, más seguros estaremos de que la muestra pertenece a la clase resultado, mientras que si el resultado de la función es próximo a 0, significa que la muestra se encuentra cerca del hiperplano y no tendremos fiabilidad de que la clasificación es correcta.

- Clasificación Multiclase.

En cuanto a la clasificación para más de dos clases $k > 2$ se pueden seguir varios métodos, siendo los más populares *One Versus One* y *One Versus All* [45].

La clasificación **One Versus One** consiste en construir $\binom{k}{2} = \frac{k(k-1)}{2}$ SVMs en total, con cada SVM haciendo comparaciones entre dos clases. Para una muestra a clasificar, se compara con cada SVM, y se realiza conteo de las veces que la muestra es clasificada a cada una de las k clases. La clase cuyo conteo es el mayor corresponde al resultado de la clasificación de la muestra.

La clasificación **One Versus All** consiste en tener k SVMs, uno por clase, comparándo la muestra a clasificar con el resto. La comparación se realiza resolviendo la fórmula general, y el resultado indica a cuánta distancia se encuentra del hiperplano. La distancia máxima de todas será indicativo de máxima fiabilidad, con lo que se le asignará esa clase.

Comenzamos entrenándola con las características y las clases del conjunto de entrenamiento, utilizando una estrategia *One VS All* y un *Gradiente Estocástico Descendiente*, el cual mejora el tiempo computacional, reemplazando el gradiente calculado usando el dataset a uno estimado calculado con muestras aleatorias del dataset. Una vez entrenado, clasificamos las características de test con el clasificador obtenido en la fase de entrenamiento y obtenemos las clases predichas. Este proceso lo repetimos por cada capa a estudiar de la CNN.

3.6. Comparación de Resultados.

Una vez hemos obtenido los resultados, componemos una matriz de confusión con las clases de las muestras de test (al ser aprendizaje supervisado contamos con ellas) y las obtenidas en el proceso de clasificación. Finalmente, obtenemos el porcentaje de acierto del clasificador convirtiendo a porcentajes los valores de la matriz de confusión y haciendo una media sobre sus valores. Realizamos el mismo procedimiento para todas las capas a estudiar de la CNN. Cuando obtenemos los resultados de la clasificación por cada capa, representamos los resultados gráficamente para compararlos y ver su eficacia.

4

Experimentos Realizados y Resultados

4.1. Descripción de los experimentos.

Los experimentos realizados cubren un análisis de las capas de la red preentrenada y el comportamiento de la extracción de características de la red según las capas que de las que se extraigan y su calidad al clasificar las características con una SVM.

Definiremos la red que utilizamos para la extracción de características, una visualización de las capas y una explicación, conjuntamente con una definición del algoritmo SVM que utilizamos para hacer la clasificación de las características, un análisis de los resultados obtenidos de los experimentos realizados, tanto con segmentación como sin segmentación. Finalmente se hará una discusión de los experimentos generales.

4.1.1. Extracción de Características

En cuanto a extraer características a partir de imágenes existen multitud de opciones utilizando *Deep Learning*, se ha optado por utilizar un tipo de ellas, las *Redes Neuronales Residuales*.

Utilizamos una ResNet-50 preentrenada para extraer nuestras características, a la cual primero le pasamos nuestro conjunto de entrenamiento generado aleatoriamente para obtener las características de entrenamiento y después, nuestro conjunto de test, para obtener las características de test.

Las características son extraídas de capas situadas al final de los bloques residuales, a distinta profundidad de la red, con la intención de ver el comportamiento de las características básicas y complejas.

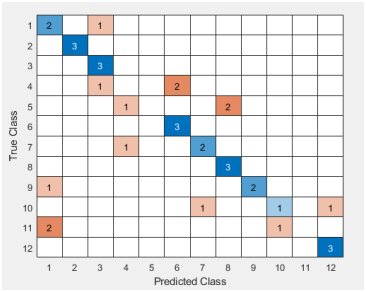
4.1.2. Clasificación de Características con SVM

Para clasificar las características obtenidas con la *Red Neuronal Residual* hacemos uso de un algoritmo de *Machine Learning*, una *SVM*.

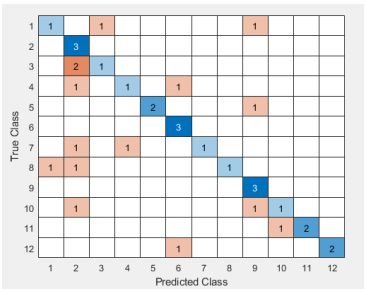
Obtenidas las características de entrenamiento mediante la *ResNet* y la etiqueta correspondiente a la muestra, entrenamos la *SVM*, donde seguimos una estrategia *One Versus All*. En el

caso de la predicción, conseguimos las características de la red entrenada y las introducimos en la *SVM* ya entrenada también, arrojándonos una serie de resultados, de los cuales medimos su eficacia con una matriz de confusión. La extracción de características y su posterior clasificación la realizamos para el dataset segmentado y sin segmentar.

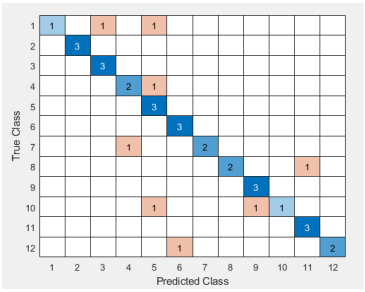
Resulta interesante realizar una clasificación de las diferentes capas de la red, a menor y mayor profundidad, tal y como se ha descrito anteriormente en los experimentos, con el fin de comparar su rendimiento en cuanto a la clasificación y poder hacernos una idea de la potencia que pueden tener características mas básicas o más complejas. Es por ello que para las mismas capas de las cuales hemos mostrado una visualización de sus características realizamos extracción y clasificación, cuyos resultados representamos de manera gráfica mediante matrices de confusión en la Figura 4.1 para las pruebas con el dataset sin segmentar y la Figura 4.2 para las pruebas con el dataset segmentado, con el fin de comprobar si el fondo, en este caso, es significativo.



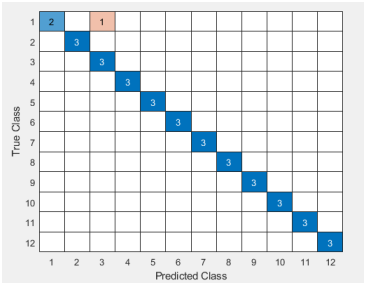
(a) Capa add_1



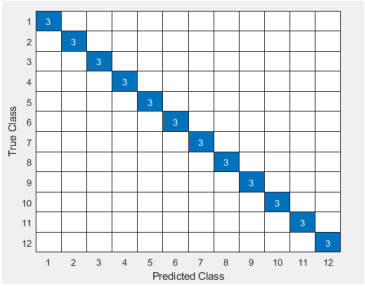
(b) Capa add_6



(c) Capa add_10

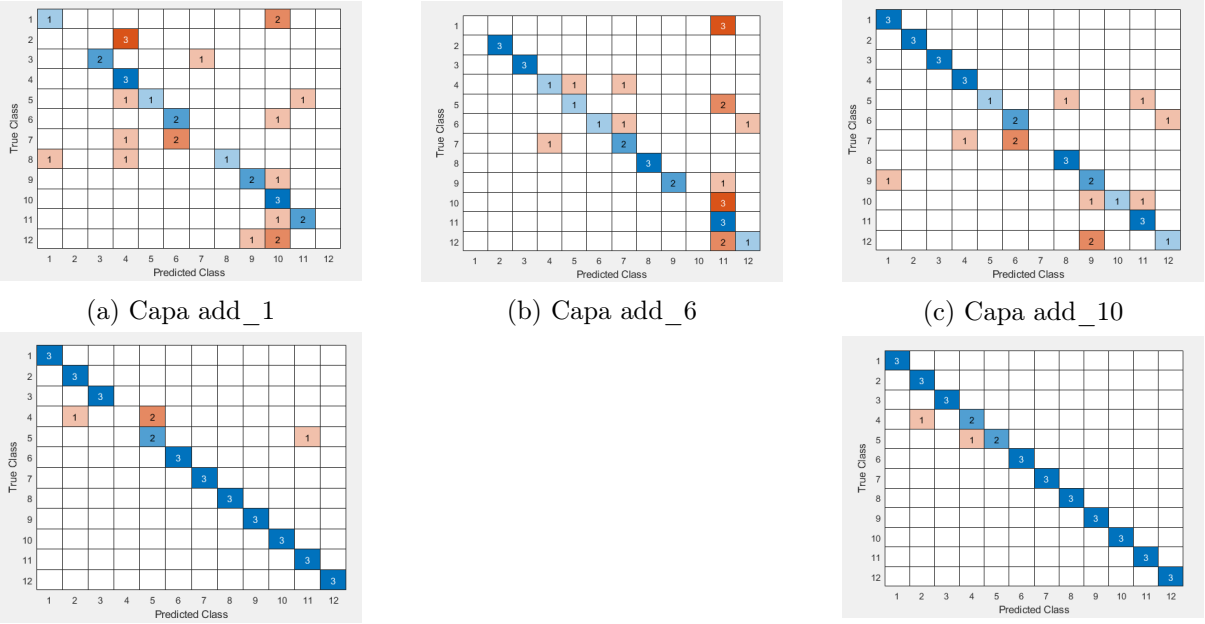


(d) Capa add_16



(e) Capa fc1000

Figura 4.1: Matrices de Confusion del Dataset sin Segmentar



(a) Capa add_1 (b) Capa add_6 (c) Capa add_10

(d) Capa add_16 (e) Capa fc1000

Figura 4.2: Matrices de Confusion del Dataset Segmentado

Con ellas nos podemos hacer una idea de la calidad de la clasificación según cada capa, así como analizar si existe una clase que es más propensa a confundirse con otra. El eje X sería la clase que el clasificador ha elegido y el eje Y es la clase que realmente le corresponde.

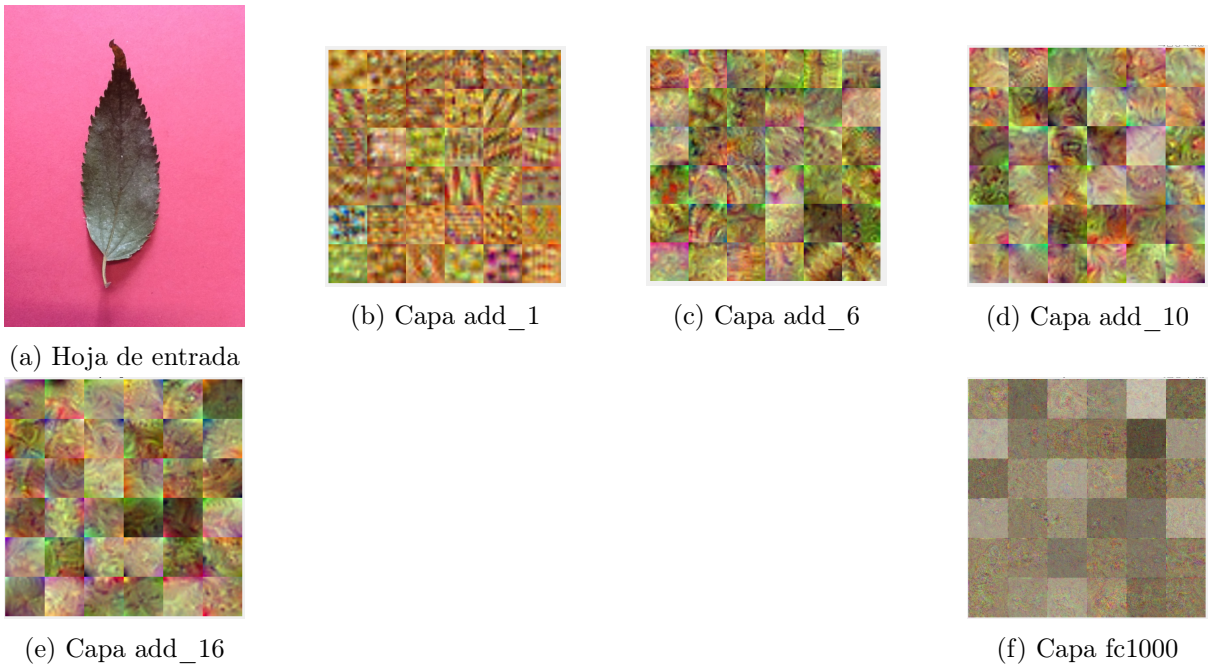
4.1.3. Análisis cualitativo de los experimentos

Las capas de la Red Neuronal pueden ser visualizadas con el fin de hacernos una idea de las características que pueden extraer, donde cuanto más compleja es la visualización que arrojan, más complejas son las características que perciben. Se incluyen una visualización de las características que detectan las capas.

Visualización de las Capas

Resulta interesante conocer el funcionamiento de la red y ver cómo se comporta según el nivel de profundidad. Se expone de una manera gráfica, como puede verse en la figura mostrando la respuesta de la red a una imagen de entrada (ver Figura 4.3) donde se ve que para una imagen de entrada, en este caso una hoja (ver Figura 4.3a), en la primera capa ilustrada la detección de características captura aquellas más simples, como el color, y a medida que se avanza consigue detectar formas más complejas y composiciones, rangos de color, texturas de la hoja, etc.

Las capas elegidas para los experimentos de extracción de características son *Addition Layers (add)* del principio, mitad y final de la red las cuales nos dan el resultado de los bloques residuales y una *Fully Connected Layer (fc1000)*, una capa totalmente conectada previa a las dos últimas capas de la red.



(a) Hoja de entrada (b) Capa add_1 (c) Capa add_6 (d) Capa add_10

(e) Capa add_16 (f) Capa fc1000

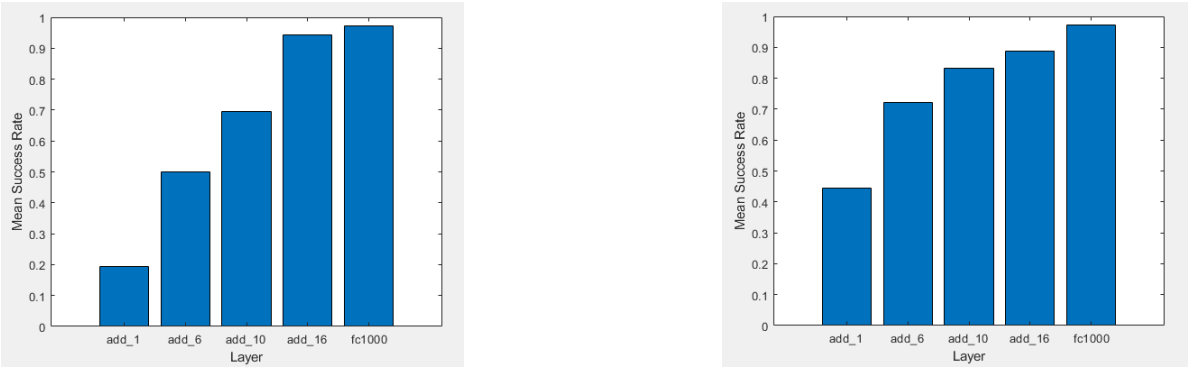
Figura 4.3: Visualización de las capas

4.2. Resultados y Discusión.

Los resultados se obtienen calculando el porcentaje de acierto a partir de la matriz de confusión, donde la diagonal representa el acierto por cada clase. Lo calculamos para cada matriz de confusión de cada capa.

Lanzamos dos pruebas, dado que la selección del conjunto de entrenamiento y test se hace al azar.

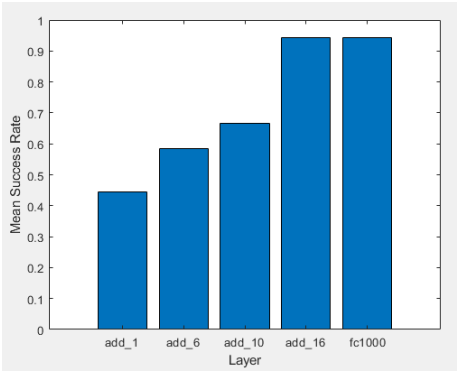
Para la clasificación **sin segmentación** obtenemos las siguientes gráficas que se aprecian en la Figura 4.4:



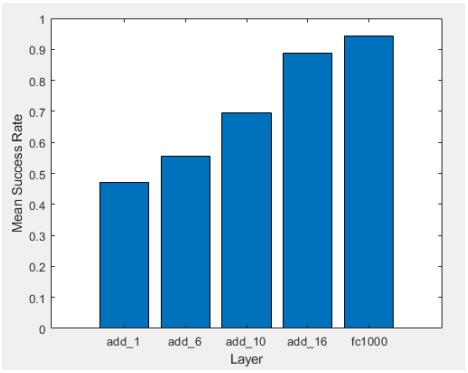
(a) Resultados de la Primera Prueba (b) Resultados de la Segunda Prueba

Figura 4.4: Porcentajes de Acierto de las Pruebas con el Dataset sin Segmentar

Para la clasificación **con segmentación** obtenemos los siguientes resultados que se ven en la Figura 4.5:



(a) Resultados de la Primera Prueba



(b) Resultados de la Segunda Prueba

Figura 4.5: Resultados de las Pruebas con el Dataset Segmentado

Las primeras capas obtienen un porcentaje de acierto pobre, pues las características que arrojan son sobretodo de color y formas básicas, cosa que en el ámbito de la clasificación de hojas, no es del todo significativo. A medida que profundizamos en la red, las características comienzan a ser más complejas, de forma, de bordes, de la textura de la hoja... Es por ello que las capas *add_16* y *fc1000* obtienen los mejores resultados, siguiendo por lo general en las pruebas lanzadas una tendencia lineal ascendente. No es uniforme debido a que el dataset de entrenamiento, al cogerlo de manera aleatoria, puede tener una peor elección de las muestras y que no resulte suficiente.

Los resultados obtenidos con las capas finales arrojan un buen porcentaje de acierto, siendo mayores de un 95 %, sabiendo que la elección de los conjuntos de entrenamiento y test es aleatoria, siendo el dataset limitado como es y teniendo en cuenta que la hoja se encuentra sobre un fondo de tonos rosas, siendo colores complementarios. También es importante decir que la red estaba preentrenada, siendo capaz ya de por sí de capturar múltiples características.

Por una parte, si el dataset hubiera sido más amplio, la red se habría especializado más, con lo que las características obtenidas serían más específicas y obtendríamos mejor resultado.

Por otro lado, el si las imágenes del dataset no hubieran tenido un fondo plano, o bien se tendría que haber realizado un preprocesamiento más intensivo, ya sea centrar la hoja para que la red entienda que se deben sacar las características del centro, discriminando el fondo, o una segmentación semántica, por ejemplo, para eliminar el fondo.

La segmentación llevada a cabo en el dataset no ha arrojado una mejora en sus resultados comparándolos con los del dataset segmentado, dado que el fondo de la imagen era un rango de color rosa, la red no lo ha considerado una característica significativa, obteniendo de igual manera características de la forma de la hoja.

También, en un escenario ideal con un dataset más grande, en un principio entrenándola con un conjunto de datos fijos y más tarde añadiendo datos nuevos al dataset, se podría conseguir una clasificación consistente.

Las redes preentrenadas dan un rendimiento notable dado un dataset limitado, pues ya son capaces de entender múltiples características, pero puede ser interesante tanto aumentar el tamaño del dataset y o bien utilizar la misma red y comprobar si mejora o utilizar una red no preentrenada por si precisara en el dataset.

5 Conclusiones

5.1. Conclusiones.

Al plantearme este estudio no tenía conocimiento sobre la base teórica de Visión por Computador y Redes Neuronales Convolucionales para caracterización ni tampoco de cómo se llevaba a cabo la identificación de hojas en Botánica, por ello hubo una previa investigación sobre la base teórica en cuanto a cómo se hacía en ellos.

La temática del proyecto implicaba disponer de un dataset normalizado de imágenes de hojas. Se realizó una investigación de bases de datos públicas que cumplieran con nuestros requisitos y se eligió una. La otra opción hubiera sido la recolección recolección y fotografiado, con todo lo que implicaba.

Se construyó un sistema de clasificación supervisada, usando para caracterización una Red Neuronal Residual y para clasificación una SVM, para ver el potencial de clasificación de características con un algoritmo de Aprendizaje Automático. Se llevó a cabo realizando la extracción de características en diferentes puntos de la red, con el fin de comparar los resultados arrojados, además de una visualización del contenido de las capas, lo cual ayudó a entender la calidad de la caracterización. El dataset se sometió a un preprocesamiento consistiendo en eliminar el fondo de la imagen. Con esto aprendí sobre preprocesamiento de imágenes, la arquitectura y funcionamiento de Redes Neuronales Residuales, la clasificación de sus características mediante algoritmos de Aprendizaje Automático y el análisis de los resultados.

Los resultados obtenidos han presentado una buena tasa de acierto en las últimas capas, mayores de 95 % en los experimentos realizados, siendo prácticamente iguales los del dataset segmentado al del dataset sin segmentar. Es gracias a que la red está pre-entrada que los consigamos dado nuestro limitado dataset, y el dataset facilita también la caracterización debido a la normalización de sus imágenes. La visualización de capas nos ha facilitado el entendimiento de cómo funciona la extracción de características y cómo se comporta a medida que profundiza en la red.

Bibliografía

[1] *https://www.docsity.com/es/clave-dicotomica-grado-de-biologia-metodos-en-biologia/3591968/.*

[2] *https://mc.ai/the-computer-vision-pipeline-part-4-feature-extraction/.*

[3] Adhi Susanto Paulus Insap Santosa Abdul Kadir, Lukito Edi Nugroho. Leaf classification using shape, color, and texture features. *International Journal of Computer Trends and Technology*, pages 1–6, 2011.

[4] Kai F. Müllerb Volker Steinhagea Pierre Barréa, Ben C. Stöverb. Leafnet: A computer vision system for automatic plant species identification. *Elsevier*, 2017.

[5] *http://alojamientos.us.es/gtocoma/pid/tema4.pdf.*

[6] *https://medium.com/@bootcampai/redes-neuronales-convolucionales-5e0ce960caf8.*

[7] *https://www.researchgate.net/figure/Image-convolution-with-an-input-image-of-size-7-7-and-a-filter-kernel-of-size-3-3$_fig1_3$18849314.*

[8] *https://www.researchgate.net/figure/Toy-example-illustrating-the-drawbacks-of-max-pooling-and-average-pooling$_fig2_3$00020038.*

[9] Fernando Berzal. *https://elvex.ugr.es/decsai/computational-intelligence/slides/N2 %20Backpropagation.pdf.*

[10] Pedro F. B. Silva and Portugal AndrÃ© R. S. Universidad de Oporto. *http://archive.ics.uci.edu/ml/datasets/Leaf.*

[11] *http://personal.cimat.mx:8181/ mrivera/cursos/aprendizaje$_p$rofundo/resnet/resnet.html.*

[12] *https://dle.rae.es/filogenia.*

[13] *https://www.ecured.cu/Taxonom %C3 %ADa.*

[14] *https://www.greelane.com/es/ciencia-tecnolog*

[15] http://librodigital.oupe.es/oxed/alumno/ciencias-de-la-naturaleza-1-eso-mec-proyecto-adarve/ebook/data/pdf/730-procedimiento10.pdf.

[16] Bruno López Takeyas. Introducción a la inteligencia artificial. pages 1–3, 2007.

[17] Geoffrey Hinton Yann LeCun, Yoshua Bengio. Deep learning. Nature 521, 436-444 (2015) doi:10.1038/nature14539*, pages 1–3, 2015.*

[18] https://towardsdatascience.com/supervised-vs-unsupervised-learning-14f68e32ea8d.

[19] D. Lu Q. Weng. A survey of image classification methods and techniques for improving classification performance. International Journal of Remote Sensing*, pages 2–6, 2007.*

[20] Gonzaled Woods. Preproceso: mejora de la imagen. Gonzaled Woods Digital Image Processing cap4.

[21] Garrison W. Cottrell Christopher Kanan. Color-to-grayscale: Does the method matter in image recognition? https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0029740*, 2011.*

[22] https://www.alphauniverse-latin.com/notas/que-es-el-contraste-y-como-puedo-aprovecharlo-en-mis-fotos.

[23] M. C. José Jaime Esqueda Elizondo. Fundamentos de procesamiento de imágenes. CONATEC 2002*, 2002.*

[24] Telecomunicación y Automática. Universidad de Jaen Departamento de Ingeniería electrónica. Preproceso: mejora de la imagen. http://www4.ujaen.es/ satorres/practicas/practica3$_v$*c.pdf*, *page* 9.

[25] https://pt.slideshare.net/AdrinPalaciosCorella/efficient-image-processing-with-hal.

[26] https://www.pinterest.es/pin/536843218053704944/?nic$_v$1 = 1*aiJKKgsL*7002*JB*9570*arIr*2*yMcbOtnfsnXvbj*%2*F*2*xRejhDgOJdaVk*5*zUp*5*BHjQTDnP*.

[27] https://www.pinterest.co.uk/pin/429812358161310170/?nic$_v$1 = 1*a*%2*FwQCx*2*H*9*FGmN*0*hN*5*TlvlrszRxc*0*I*49*JEIVgYE*%2*FMjOs*5*VwK*1*f*%2*Bx*%2*FjrqXfoCom*

[28] https://medium.com/peritos-solutions/id-card-border-detection-using-emgu-cv-59aa1114397d.

[29] Saurabh Srivastava A. K. Sinha Rajkumar Goel, Vineet Kumar. A review of feature extraction techniques for image analysis. International Journal of Advanced Research in Computer and Communication Engineering*, pages 1–3, 2017.*

[30] Gebhard Banko. A review of assessing the accuracy of classifications of remotely sensed data and of methods including remote sensing data in forest inventory. International Institute for Applied Systems Analysis*, pages 1–5, 1998.*

[31] https://empresas.blogthinkbig.com/ml-a-tu-alcance-matriz-confusion/.

[32] https://www.kaggle.com/alexo98/leaf-det.

[33] https://www.kaggle.com/xhlulu/leafsnap-datasetleafsnap-dataset-images.txt.

[34] Neeraj KumarPeter N. BelhumeurArijit BiswasDavid W. JacobsW. John KressIda C. LopezJoão V. B. Soares. Leafsnap: A computer vision system for automatic plant species identification. European Conference on Computer Vision*, 2012.*

[35] http://flavia.sourceforge.net/.

[36] Lic. Ulises Román Concha Dra. Nora La Serna Palomino. Técnicas de Segmentación en Procesamiento Digital de Imágenes*, pages 11–12.*

[37] Patricio Loncomilla. Deep learning: Redes convolucionales. https://ccc.inaoep.mx/ pgomez/deep/presentations/2016Loncomilla.pdf.

[38] Patricio Loncomilla. Deep learning: Redes convolucionales. https://www.aprendemachinelearning.com/como-funcionan-las-convolutional-neural-networks-vision-por-ordenador/.

[39] https://relopezbriega.github.io/blog/2016/08/02/redes-neuronales-convolucionales-con-tensorflow/.

[40] https://mlearninglab.com/2018/11/04/entrenamiento-de-redes-convolucionales-y-teoria-de-la-informacion/.

[41] https://es.mathworks.com/company/newsletters/articles/a-brief-history-of-matlab.html.

[42] https://guiasjuridicas.wolterskluwer.es/Content/Documento.aspx?params=H4sIAAAAAAAEAMtMSbF1j$ckhlQaptWmJOcSoAOSON0zUAAAA = WKE$.

[43] Shaoqing Ren Jian Sun Kaiming He, Xiangyu Zhang. Deep residual learning for image recognition. Microsoft Research.

[44] Cristina Gil Martínez. https://rpubs.com/Cristina$_{G}il/SVM$, 2018.

[45] https://www.cienciadedatos.net/documentos/34$_{m}aquinas_{d}e_{v}ector_{s}oporte_{s}upport_{v}ector_{m}achines$.